User's Guide

Genesys Web Engagement 8.5.0

1/22/2022

Table of Contents

Genesys Web Engagement User's Guide

Welcome to the *Genesys Web Engagement 8.5 User's Guide*. This document provides procedures and instructions for common tasks you need to perform when setting up and configuring Genesys Web Engagement. Most of this information can be found in other GWE guides — it's compiled here as a quick reference for your convenience. See the summary of chapters below.

First Look

This should be the first thing you read when you want to understand Genesys Web Engagement.

A First Look at Genesys Web Engagement

Business Information

Learn how to manage business information.

Managing Categories

Managing Business Events

Rules

Find information to help you manage rules templates and rules packages.

Managing CEP Rules Templates

Managing Rules

Tools

Find information about how to use the Web Engagement Instrumentation Tool.

Using InTools

A First Look at Genesys Web Engagement

Watch the video!

Genesys Web Engagement tracks the online activity of your customers, because sometimes it's hard for them to figure out how to get what they want.

With the information we give you, you can often help them find their way:

As you can see, we provide many types of engagement. Let's look at an example of assisted service.

I need to buy a new HDTV, and I've heard of a website called acmetv.genesys.com that is supposed to have some great deals this week. It's time to go shopping!

This company uses a technology called Genesys Web Engagement to see where their customers have been on their website. They can tell which pages I have visited and how long I have been at each one.

When I first get to their website, as the page loads, it processes a JavaScript snippet. This snippet sets up browser tier widgets and agents that send events to the Web Engagement software.

HDTV-A

I have a particular model in mind, the HDTV-A, so I find the page for that model and start reading the reviews.

It turns out that this model isn't as popular as I thought it would be, because people just don't like the image quality. At this point, the vendor could have offered to engage with me. But I probably need to get more information before I am ready to be helped, so they wait for a while.

Product search

I have done a product search and found two models that I'm interested in. They both have a good price, but one is higher resolution and a lot larger. Here they are:

HDTV-B

HDTV-C

Comparison

I like both models, but I am spending some time looking at the pages for each one to narrow my choice down to just one.

This is another point where the company could engage with me, but maybe all of their agents are busy right now.

Anyway, I finally decide that I really like the HDTV-C, and the price is reasonable enough that I am ready to buy!

Time to engage!

I have started filling out the online form, but I am getting stuck for some reason—there is a field that I just can't get to work. And I am also slowing down a bit because maybe I don't really want to spend quite so much money right now...

Well, do you remember the JavaScript snippet? And the events it's been sending?

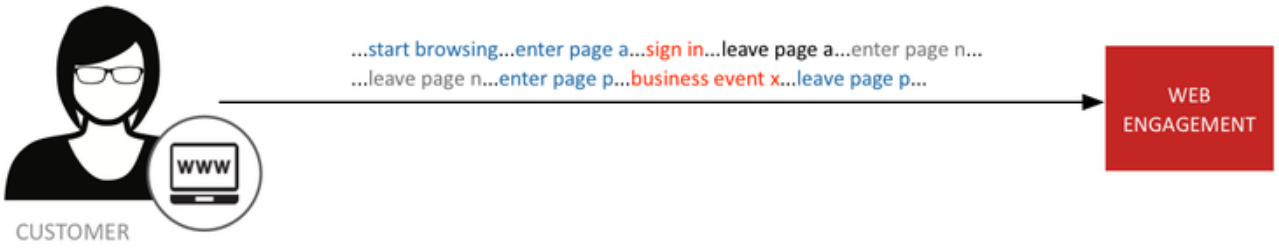
Acmetv.genesys.com has been paying attention, so they know I'm interested in the HDTV-C. And they know that some customers are abandoning after initially selecting this model. So after spending a couple of minutes on the shopping cart page, I suddenly receive a chat request!

I'm pretty glad to get some help at this point, so I respond to the request. After a brief conversation, I decide that I really do want to buy the HDTV-C. The agent helps me fill out the field I had trouble with—and my new TV will be delivered tomorrow!

That was easy!

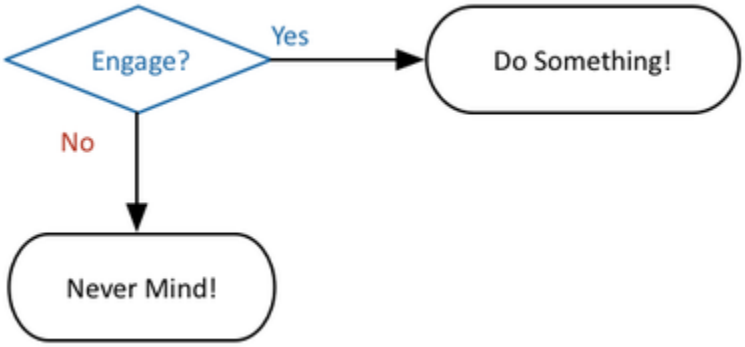
Behind the scenes

As we mentioned, Web Engagement captures events from the customer's browser. In the simplest scenarios, you can use our built-in events, which give you information about things like how long a customer has been browsing and what pages they have visited.



When the Web Engagement server receives events from the browser, it uses them as input to rules. One of the most asked-for of these rules waits until a person has been on the site or on a specific page for a certain amount of time and then decides what to do at that point.

In this case, acmetv.genesys.com knows that if I have been on the shopping cart page for more than 2 minutes, they should probably send me some help. Pretty simple!



All of this technology is based on what we call the Simple Engagement Model.

But sometimes you need to be a bit more sophisticated. As in determining whether to tell me about the HDTV-B and the HDTV-C. For that kind of situation, you can set up your own events, called Business Events. If acmetv.genesys.com was getting a lot of abandons after people gave up on the HDTV-A, and they noticed that a lot of people who did buy were going for the HDTV-C, they could have used custom business events to help determine how to engage the customers who were about to abandon.

Or they might set up special events that determine whether someone is comparing two models, so

they can help them decide.

We refer to this as the Advanced Engagement Model.

Simple Engagement Model

If you only need System Events, there are two main things you need to do:

- Set up categories for your pages. For example, maybe the HDTV-C is only one of several models that have been leading to abandoned transactions. You can set up a category for all of these models, rather than hard-coding a specific URL.
- Choose the template of the rule that suits your business needs.

Advanced Engagement Model

If you need your own custom events, you also have to create Business Events in the Domain-Specific Language (DSL) files.

Getting started

Before setting up categories, or rules, or business events, acmetv.genesys.com had to install Web Engagement. They also had to do some customization that required help from people with skills in JavaScript and jQuery (although you may not need jQuery for your installation).

Want to learn more?

For a more in-depth look at Web Engagement, read the **Product Overview**.

Watch the video!

Link to video

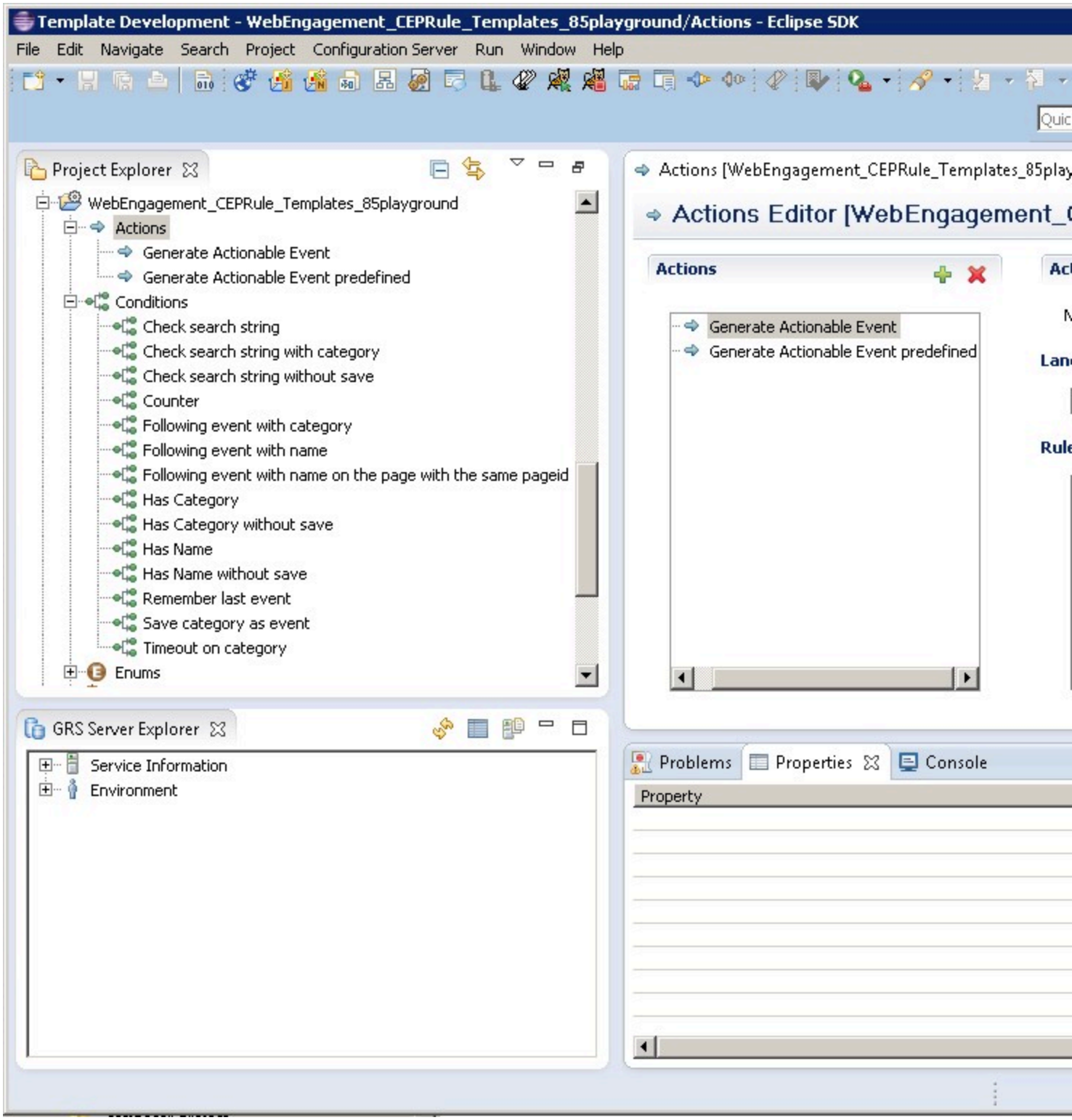
Managing CEP Rule Templates

Overview

The Complex Event Processing (CEP) Rule Templates define the actions and conditions you can use when you create your business rules in Genesys Rules Authoring Tool.

You use Genesys Rules Authoring Tool (GRAT) to develop, author, and evaluate these 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. GWE provides an out-of-the-box CEP (Complex Event Processing) Rules template. 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 out-of-the-box rule templates created with your Web Engagement application are represented as a GRDT project, which is located at **\apps*application name*\resources_composer-projects\WebEngagement_CEPRule_Templates**.



CEP rule template in Composer

In order to use these templates to define rules, you must first publish them — see Publishing the CEP

Rule Templates for details about when the templates should be published in the Web Engagement application development workflow.

Before you publish the templates, you can edit them to suit your business needs using the the Genesys Rules Development Tool. For more information about rule templates, refer to the Genesys Rules System documentation.

> **Important**
>
> Note that if you customize your rule templates, you must republish them.

Actions

The list of actions available in the template is listed in **WebEngagement_CEPRule_Templates > Actions**. You can edit, add, or remove these actions. In the Genesys Rules Authoring Tool (GRAT), when you create a rule based on the template, you can add an action by clicking **Add action**; GRAT displays all the actions defined in the template. You'll see how actions are implemented once you start creating rules. The default actions are:

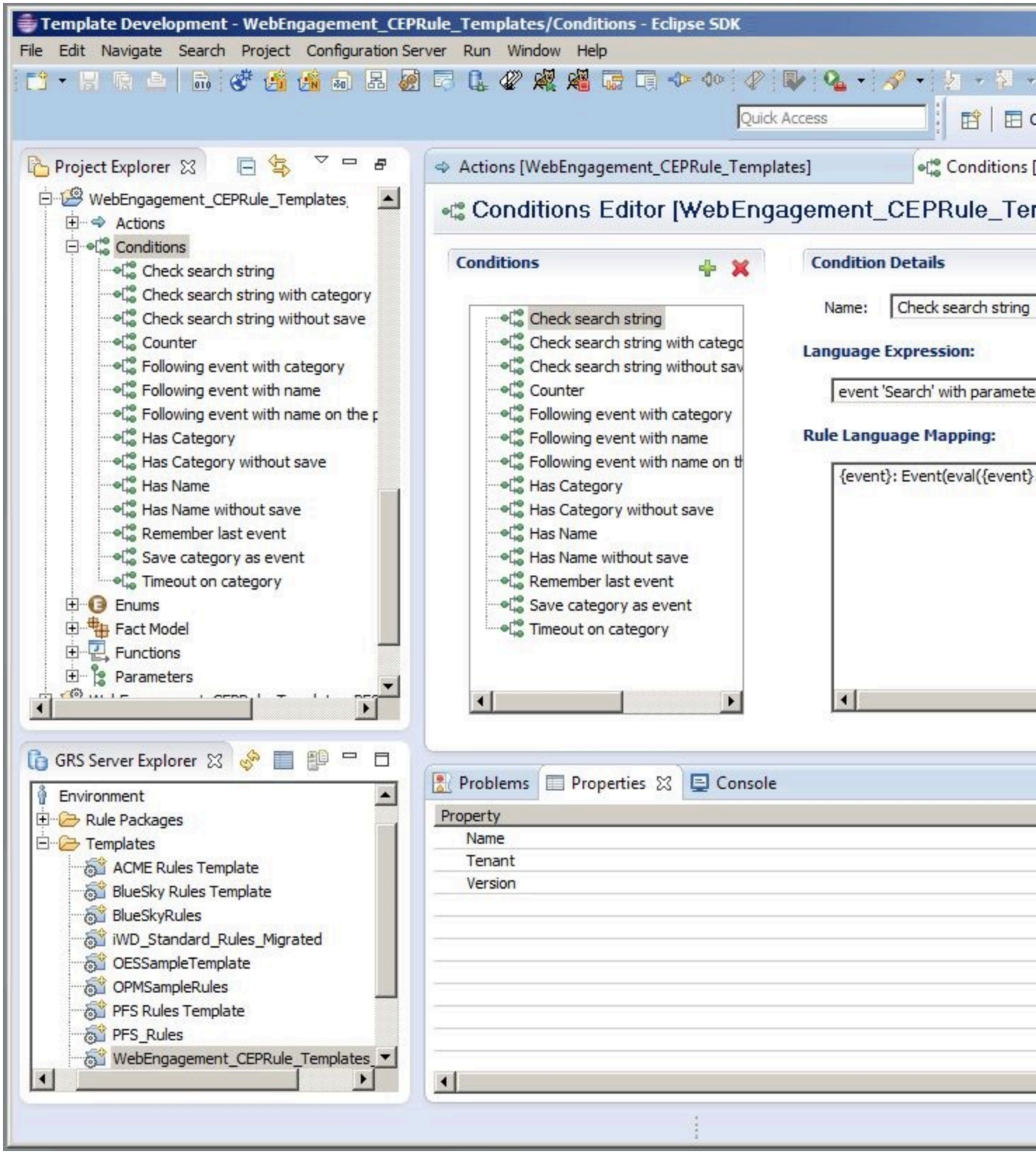
- Generate Actionable Event
- Generate Actionable Event Predefined

Enums

The enumerations available in the template are listed in **WebEngagement_CEPRule_Templates > Enums**. You can edit, add, or remove these enumerations. When you create a rule based on the template, you can specify a **Phase** by clicking **Add Linear Rule**; GRAT displays all the enumerates available in the template. In the default template, no specific enumeration is available.

Conditions

The conditions are listed in **WebEngagement_CEPRule_Templates > Conditions**.


Template Development - WebEngagement_CEPRule_Templates/Conditions - Eclipse SDK
File Edit Navigate Search Project Configuration Server Run Window Help
Quick Access
Project Explorer
WebEngagement_CEPRule_Templates
Actions
Conditions
Check search string
Check search string with category
Check search string without save
Counter
Following event with category
Following event with name
Following event with name on the p
Has Category
Has Category without save
Has Name
Has Name without save
Remember last event
Save category as event
Timeout on category
Enums
Fact Model
Functions
Parameters
Actions [WebEngagement_CEPRule_Templates]
Conditions Editor [WebEngagement_CEPRule_Ter
Conditions
Check search string
Check search string with catego
Check search string without sav
Counter
Following event with category
Following event with name
Following event with name on th
Has Category
Has Category without save
Has Name
Has Name without save
Remember last event
Save category as event
Timeout on category
Condition Details
Name:
Check search string
Language Expression:
event 'Search' with paramete
Rule Language Mapping:
{event}: Event(eval({event}
GRS Server Explorer
Environment
Rule Packages
Templates
ACME Rules Template
BlueSky Rules Template
BlueSkyRules
iWD_Standard_Rules_Migrated
OESSampleTemplate
OPMSampleRules
PFS Rules Template
PFS_Rules
WebEngagement_CEPRule_Templates
Problems
Properties
Console
Property
Name
Tenant
Version

List of conditions in the CEP rule template.

You can edit, add, or remove these conditions. Each condition associates a name with an expression. When you create a rule based on the template, you can add one or more condition to this rule by clicking **Add condition**; GRAT displays all the condition expressions available in the template. For complex templates, you need several conditions to implement a rule.

Condition Details

Condition Name	Expression	Condition details
Check search string	event searches {searchString}	Returns true if the event Search occurs and if the {searchString} label is found, this event's result is saved in the {event} label.
Following event with category	AND event following {prevEvent} with category {category} save as {event}	If the event follows {prevEvent} and contains the {category} label, this event's result is saved in the {event} label.
Following event with name	AND event following {prevEvent} with name {eventName} save as {event}	If the {eventName} follows {prevEvent} in parameter, this event's result is saved in the {event} label.
Has Category	page transition event occurs that belongs to category {category} save as {event}	If the event is a page transition for the given category, this event's result is saved in the {event} label.
Has Category without save	page transition event occurs that belongs to category {category}	Returns true if the event is a transition to the given category's page.
Has Name	event with name {eventName} save as {event}	If the {eventName} occurs, this event's result is saved in the {event} label.
Has Name without save	AND event with name {eventName}	Returns true if {eventName} occurs.
Remember last event	Precondition: save last event	Saves the last event.
Save category as event	category is {category} save as {event}	If the event contains the given category, this event's result is saved in the {event} label.
Timeout on category	Timeout event occurs with category {category}	Returns true if the Timeout event occurs for the given category.

Importing the CEP Rule Templates in GRDT

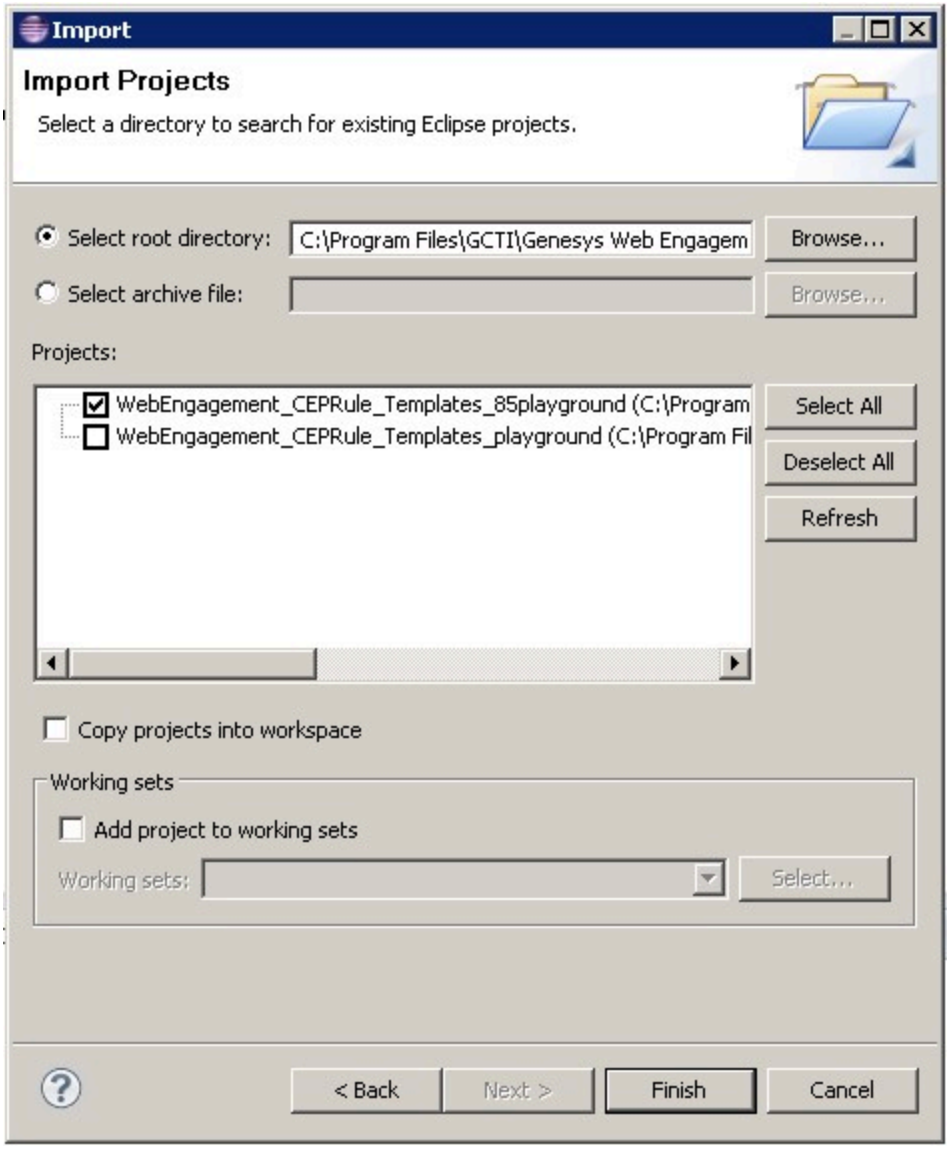
Complete this procedure to import the CEP rule templates in the Genesys Rules Development Tool. Even if you do not plan to customize the templates, your rule template must be published in the Rules System Repository before you try to create rules.

Prerequisites

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

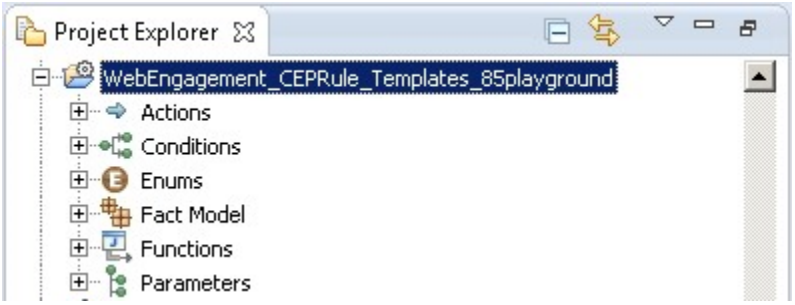
Start

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. Your newly created Genesys Web Engagement application includes a GRDT project with rules templates located at **\apps*application name*\resources_composer-projects\WebEngagement_CEPRule_Templates**. Select the rules template project to import:
   - Browse to the **\apps*application name*\resources_composer-projects** folder in the Genesys Web Engagement installation directory and select a project.
   - Click **OK**. **WebEngagement_CEPRule_Templates_*application name*** is added to the **Projects** list.
   - Select the **WebEngagement_CEPRule_Templates_*application name*** 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_*application name*** is added to the **Project Explorer**.



WebEngagement_CEP_Rule_Templates_playground is added to the Project Explorer.

End

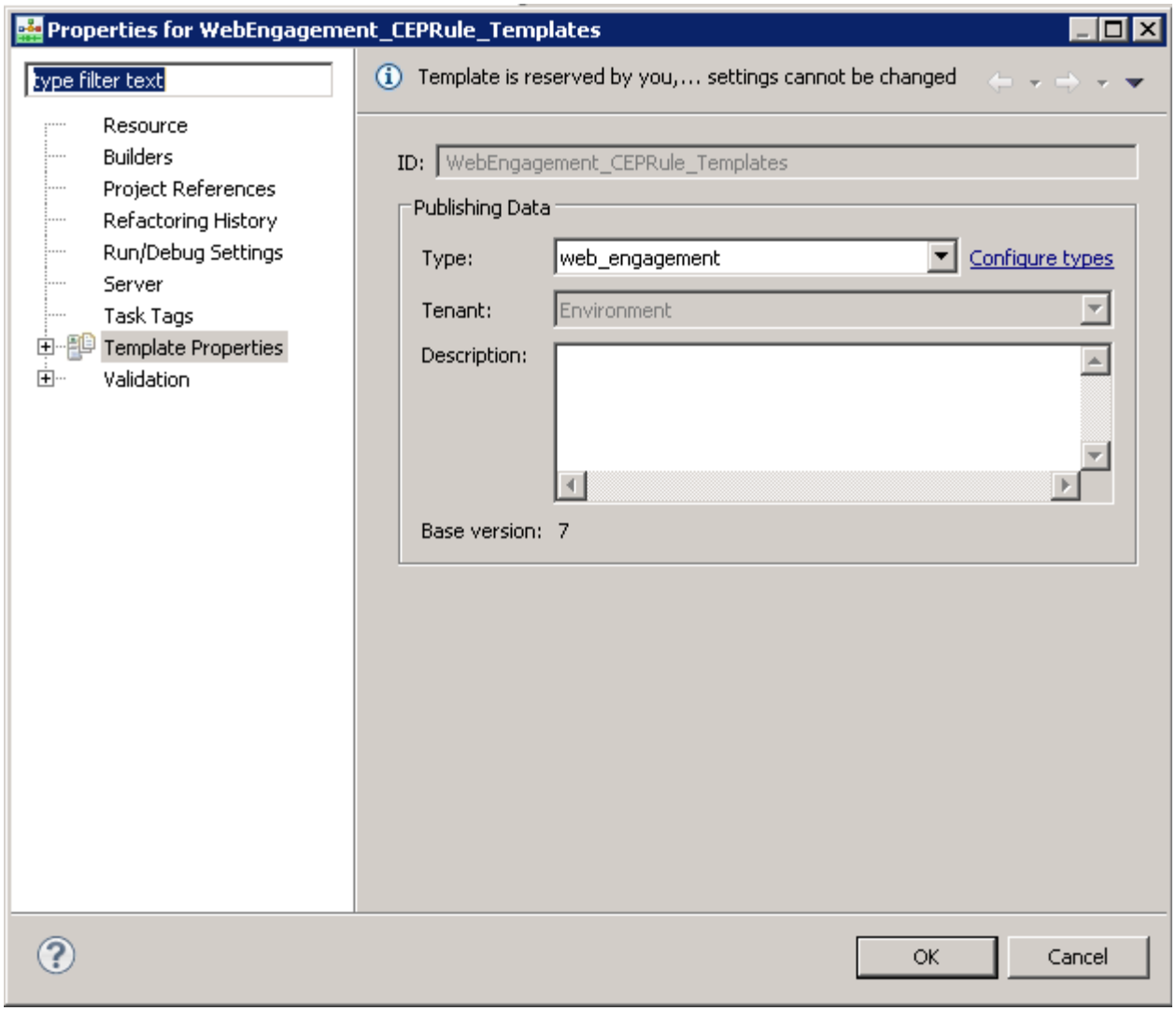
Configuring the CEP Rule Templates

Prerequisites

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

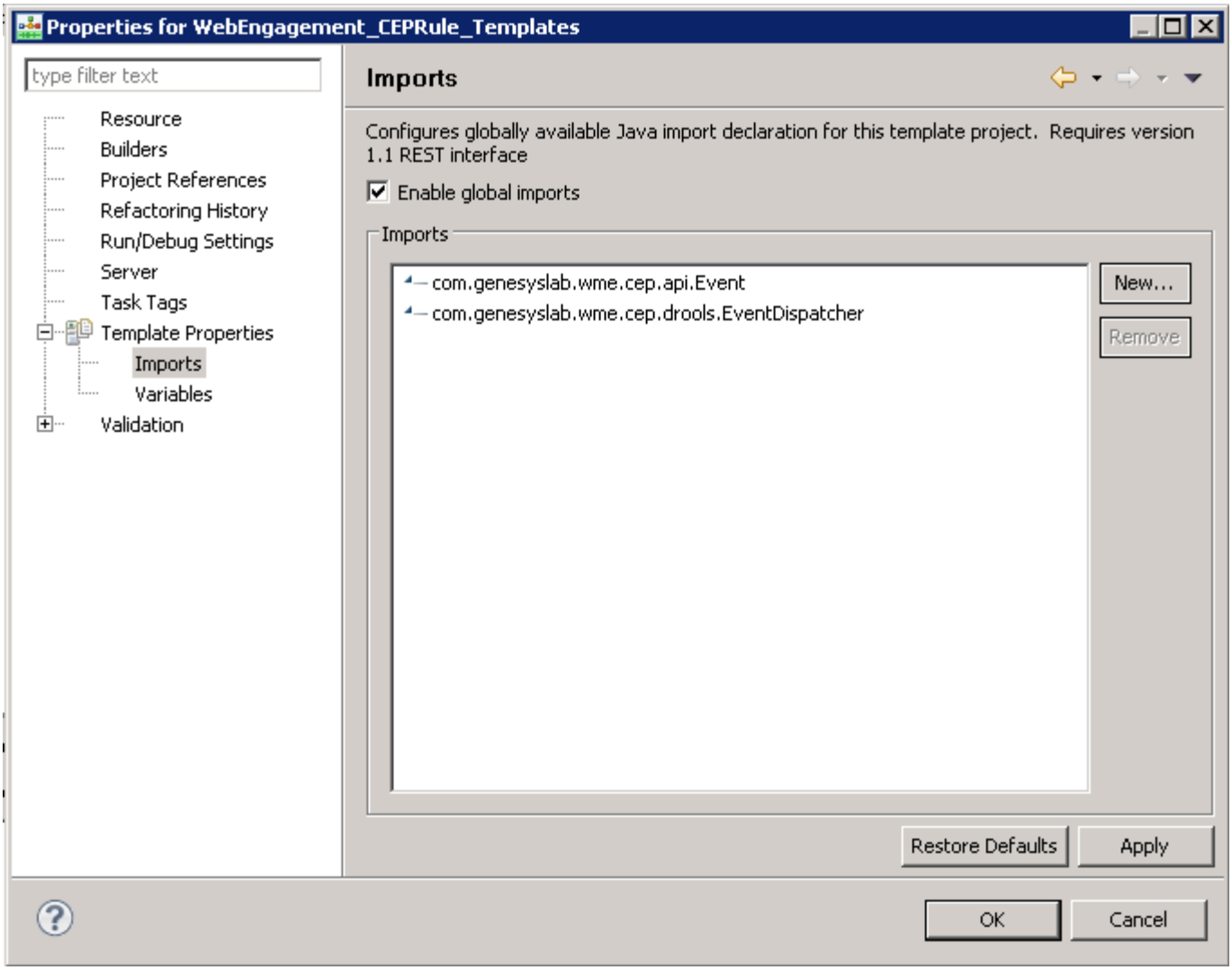
Start

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`.

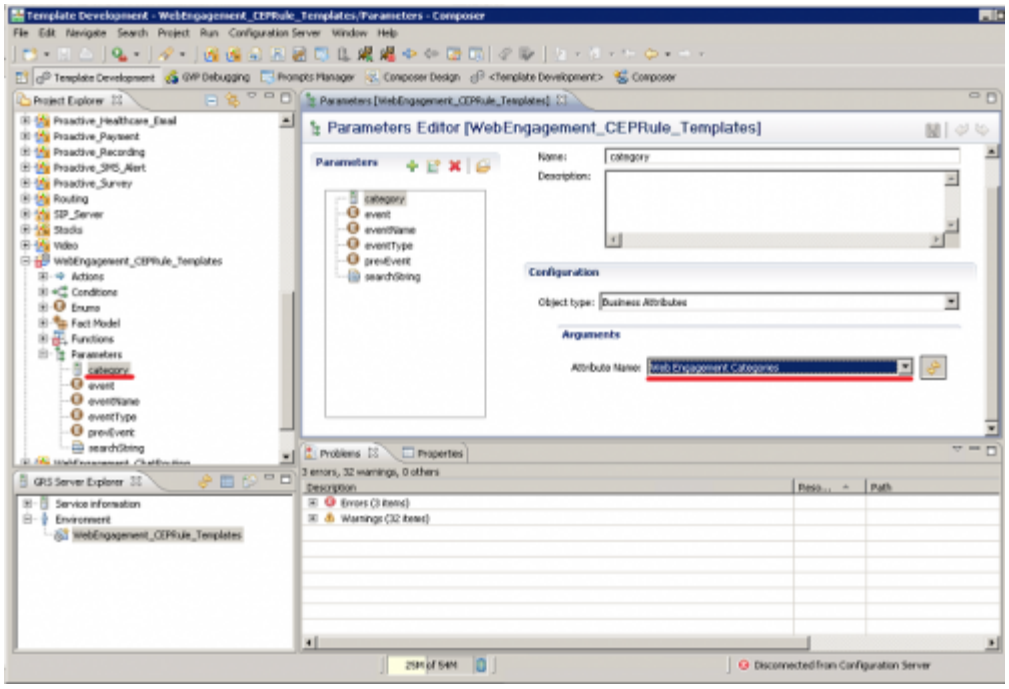
3. Navigate to **Template Properties > Imports**. The **Imports** panel opens.
4. Select the **Enable global imports** option.



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`.



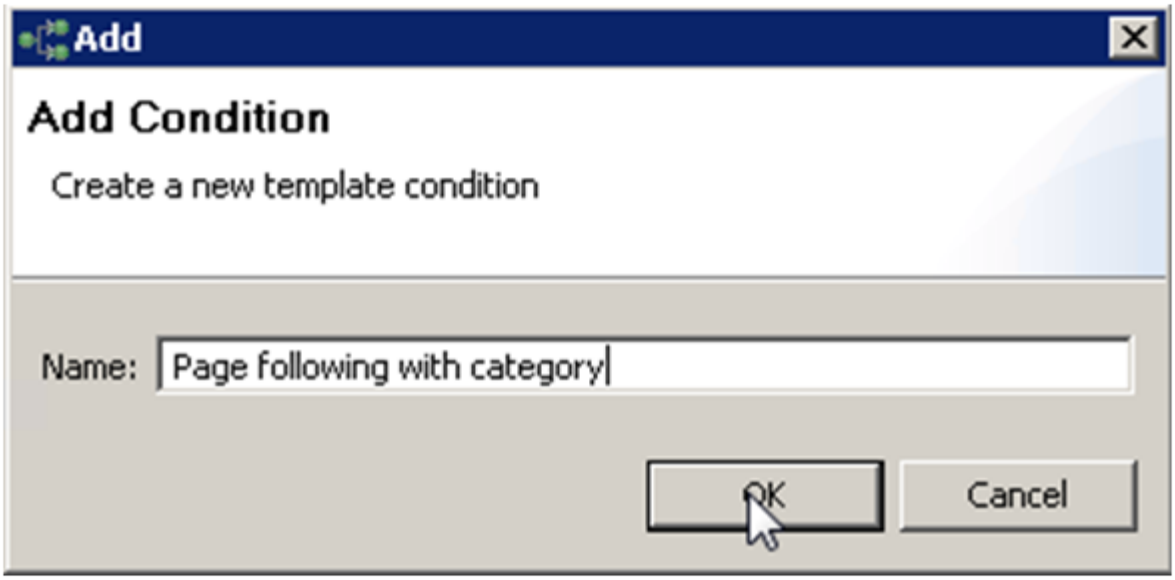
8. Click **Save**.

End

Customizing the CEP Rule Templates (Optional)

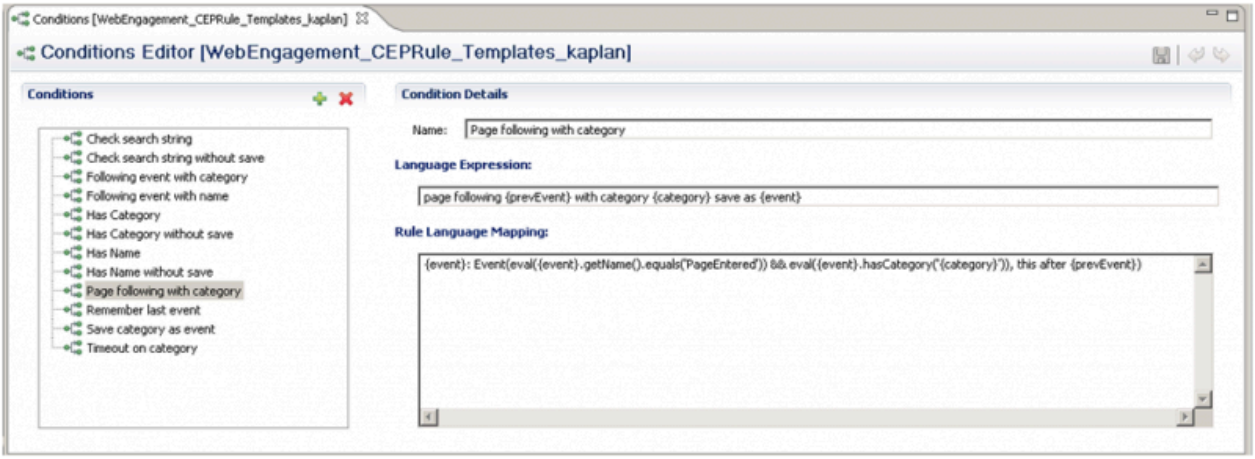
Start

1. Open the CEP rule template project with GRDT and navigate to the Conditions item.
2. Expand Conditions to open the Conditions editor.
3. In the Conditions tab, click **+**. The **Add Condition** window opens.



Add a condition

4. Enter a name and click **OK**. The condition is added and selected in the condition list; the condition detail panel opens.
5. Insert the Language Expressions and Rule Language Mapping:



6. Click Save Now when the rule template is published, the rule will be available in GRAT:

End

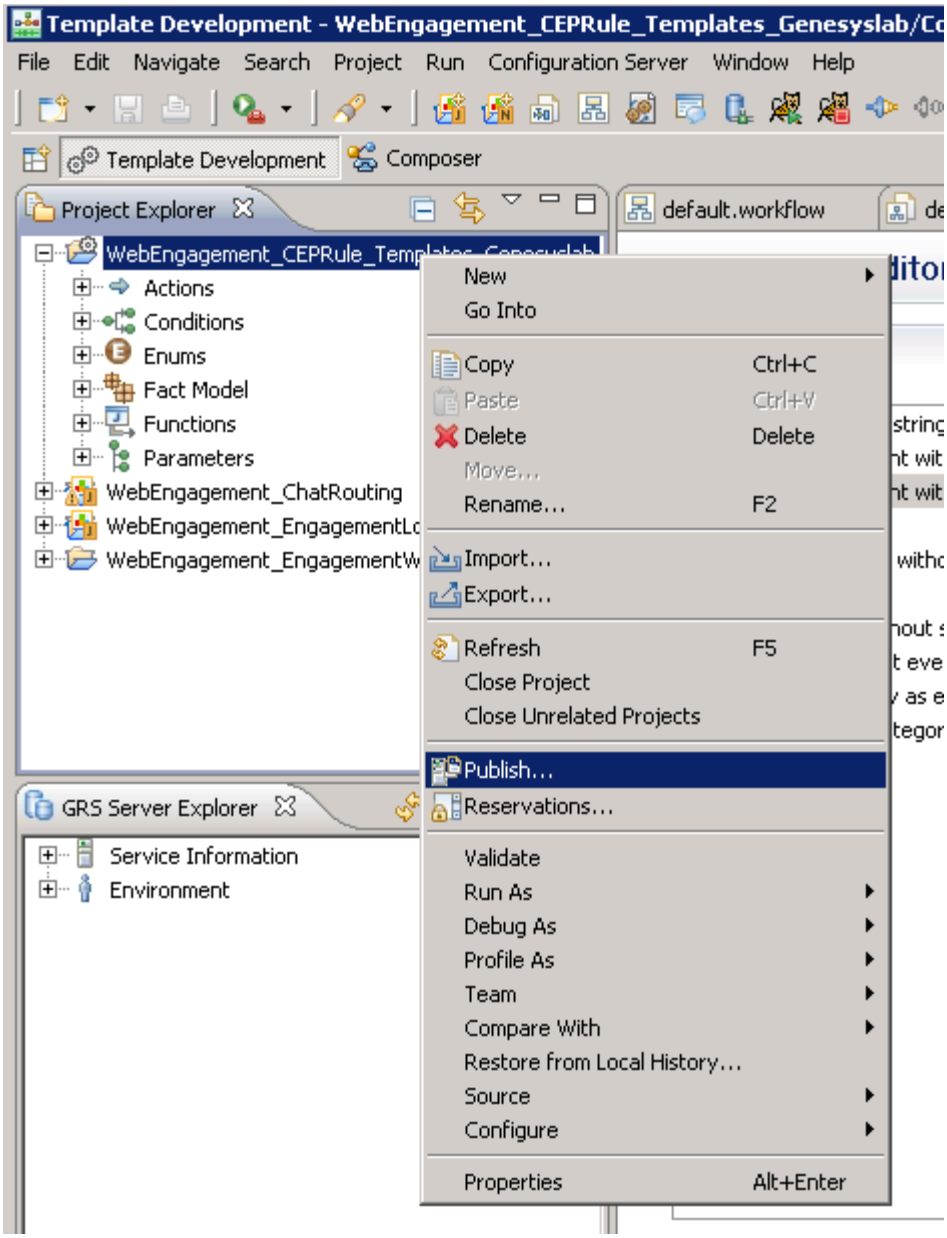
Publishing the CEP Rule Templates in the Rules Repository

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.

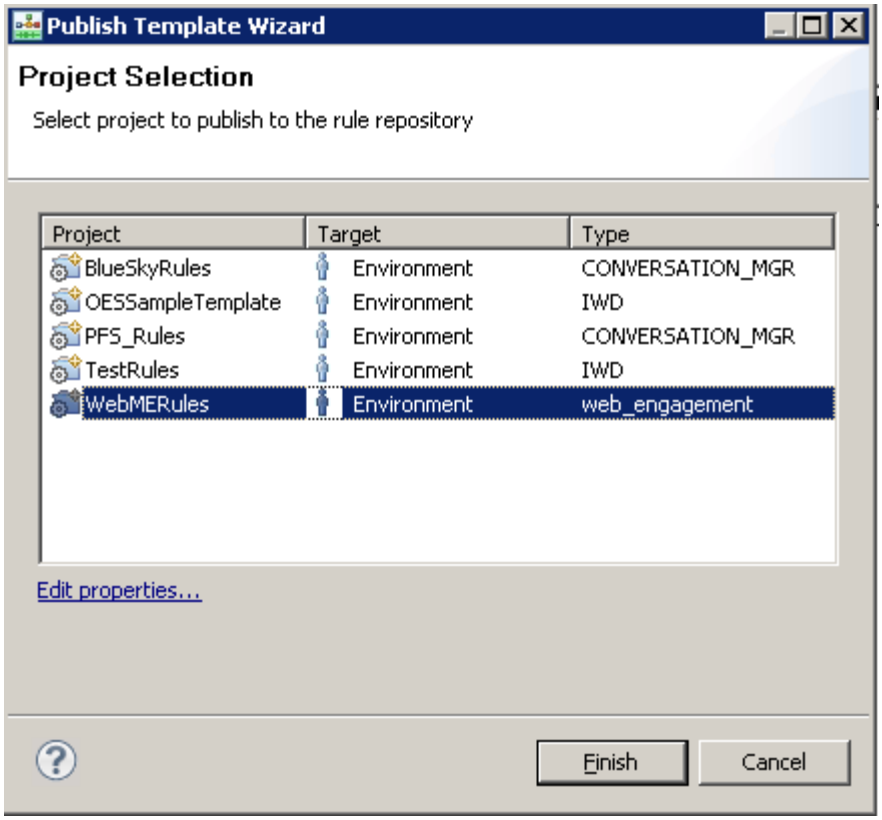
Start

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

Working with Categories

About Categories

When we think about things, or talk about them with our friends, we often categorize them. This can help us tell these things apart from other things that are similar to them, which means that these categories can guide us towards better decisions.

As it happens, Web Engagement needs to do some serious decision-making. Because of this, we have set up ways for you to categorize things within Web Engagement, so it can make those decisions for you as you try to help your customers in real time.

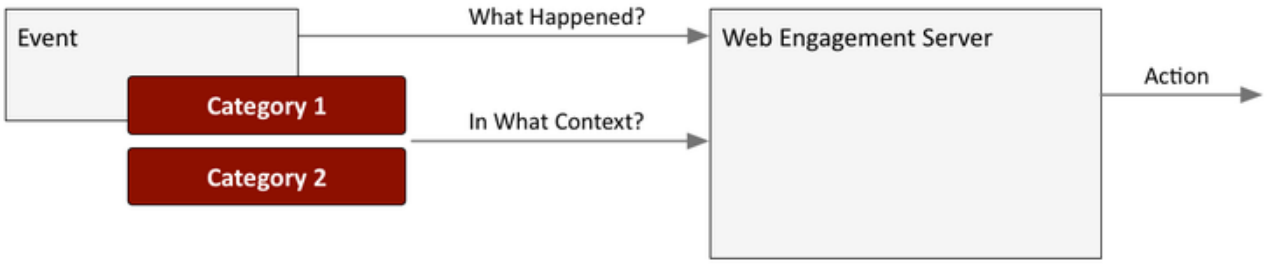
In addition to its value as a real-time analysis tool, this categorization provides a valuable way of grouping the monitoring and engagement data that goes into your reporting.

How Categories Differ from Events

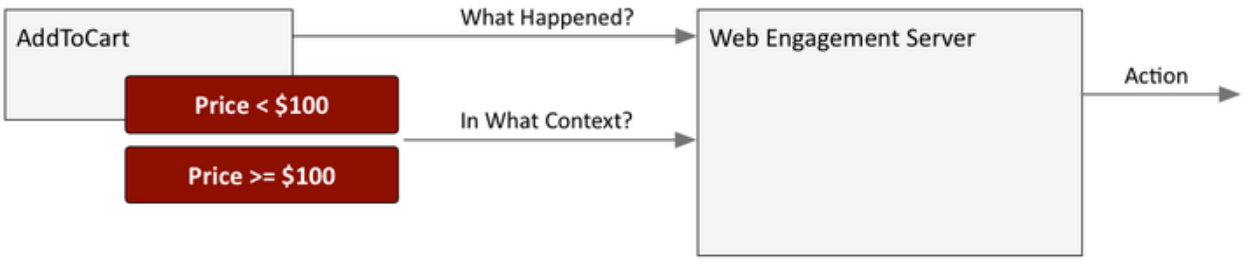
You can think of a Web Engagement category as a special kind of non-hierarchical label that is assigned to an event, based on some particular characteristic of the event's data.

You can use things like the URL, the language code, the page title—or other data passed along with the event—to figure out whether that specific event fits into a particular categorization. This allows you to combine similar events into a single customer business context so you can answer questions like *What happened?*—which you will mainly learn from the event itself—and *In what context?*—which will be clarified by the categories associated with one or more events.

By combining two sets of information like this, you get an additional layer of meaning within which to bucket your events, which allows you to simplify your rules.



In the following example, we have set up categories based on the price of an item added to the user's cart. We might want to send engagement invitations to anyone who is buying the higher-priced items, while being more selective towards people with lower-end transactions.



Another big benefit of categories is that they are really easy to configure—all you have to do is open the Web Engagement GAX plugin and create the categories, and they are ready to use!

How Categories Make Events More Useful

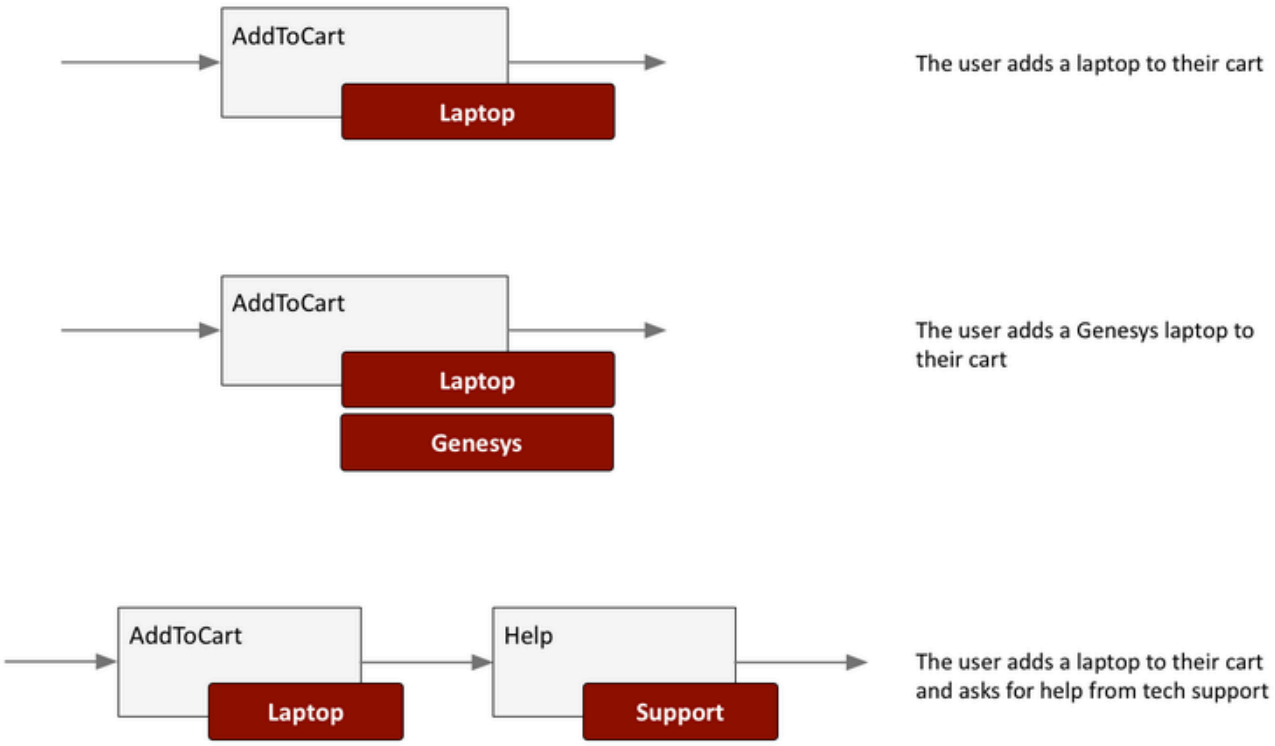
Much of the time, the name and type of an event are enough to tell you *What happened?*

For example, an **AddToCart** event lets you know that the customer has added something to their shopping cart. But if you only rely on this, you'll have to create specific events for a lot of different actions if you want to get a good idea of what your customers are doing.

By assigning categories, you add another dimension to how you can group events, thereby cutting down on the number of events you need to set up if you want to understand your most important business criteria.

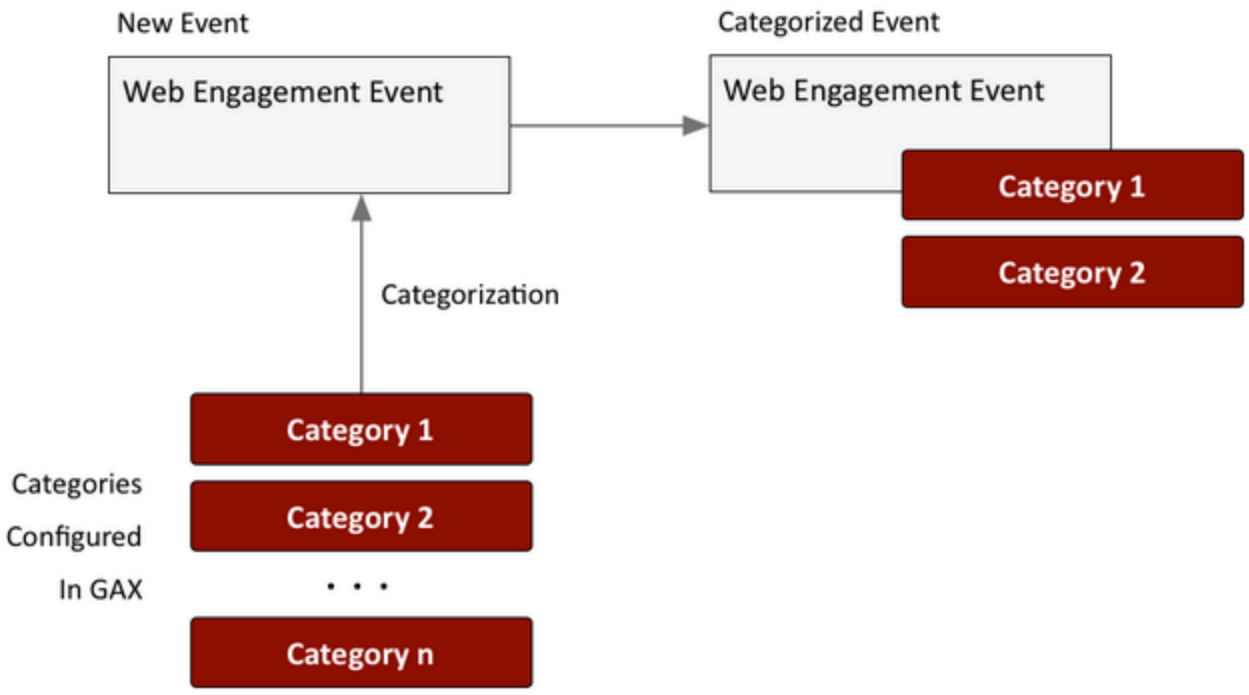
For example, in the case of an online store, an **AddToCart** event can be categorized by what kind of thing the customer wants to buy, such as a **Phone** or a **TV** or a **Laptop**, or by other important characteristics, such as the price of the item. You can also add categories that let you know when a customer is **Waiting For Support**, or wants to **Stop Shopping**.

Any and all of these categories can be a factor in triggering actions that generate a proactive engagement request—a request which could make the difference between a sale and an abandonment.



Assigning Categories to New Events

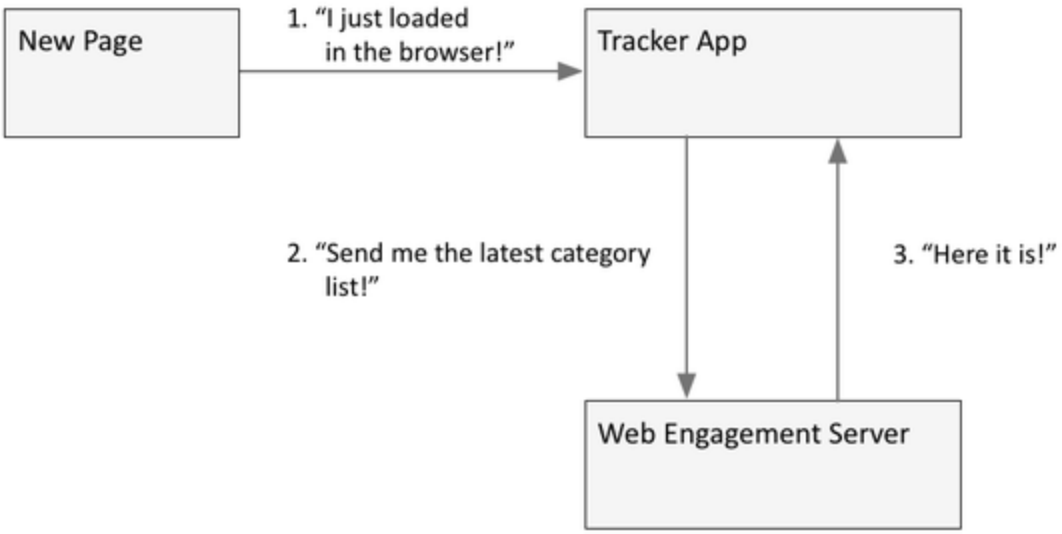
Every time the Genesys Tracker application—which is resident in the visitor's browser—generates an event, it tries to categorize it based on the category information received from the Web Engagement Server.



Workflow

Let's take a look at how these things come together when a customer is visiting your online store.

The Genesys Tracker application is constantly monitoring the pages on your website. As soon as a new page is loaded in the customer's browser—as shown in **Step 1**, below—the Tracker application asks for the current list of categories from Web Engagement Server (**Step 2**), which sends it to the Tracker (**Step 3**).



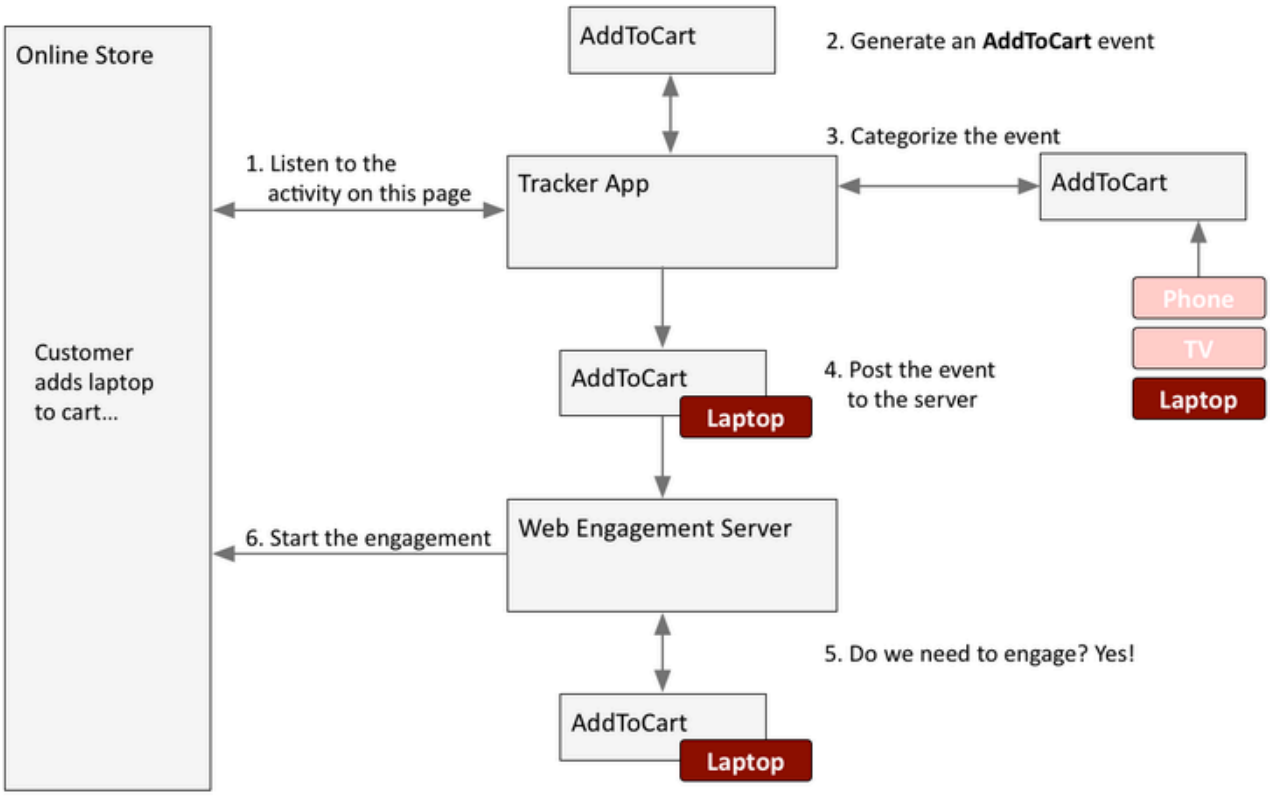
Here's what happens when a customer adds a laptop on an online store page, as shown in the following diagram.

Tracker is still monitoring the events associated with the new page, as shown in **Step 1**. When the customer adds the item to their cart, the Tracker app creates an **AddToCart** event (**Step 2**) and categorizes it by purchase category (**Step 3**). (Notice that the event is categorized on the browser side.)

The Tracker app then posts the categorized event to the Web Engagement Server (**Step 4**). At this point (**Step 5**), Web Engagement Server can apply its rules to all of the information that is available from the event, in order to determine whether to engage with the customer. This information can include:

- The event name and type
- The specific combination of categories that were applied to the event
- The specific combination of events and categories associated with the recent customer activity

The outcome from this activity could be a decision to provide assistance via chat, callback, or other channels (**Step 6**), or the server could simply decide to let the Tracker app continue monitoring.



Simple and Advanced Engagement Models

Web Engagement supports two types of events. Out of the box, you have access to several different system events, which are used in the Simple Engagement Model. You can also create your own events, known as business events, in which case you are using what we call the Advanced Engagement Model.

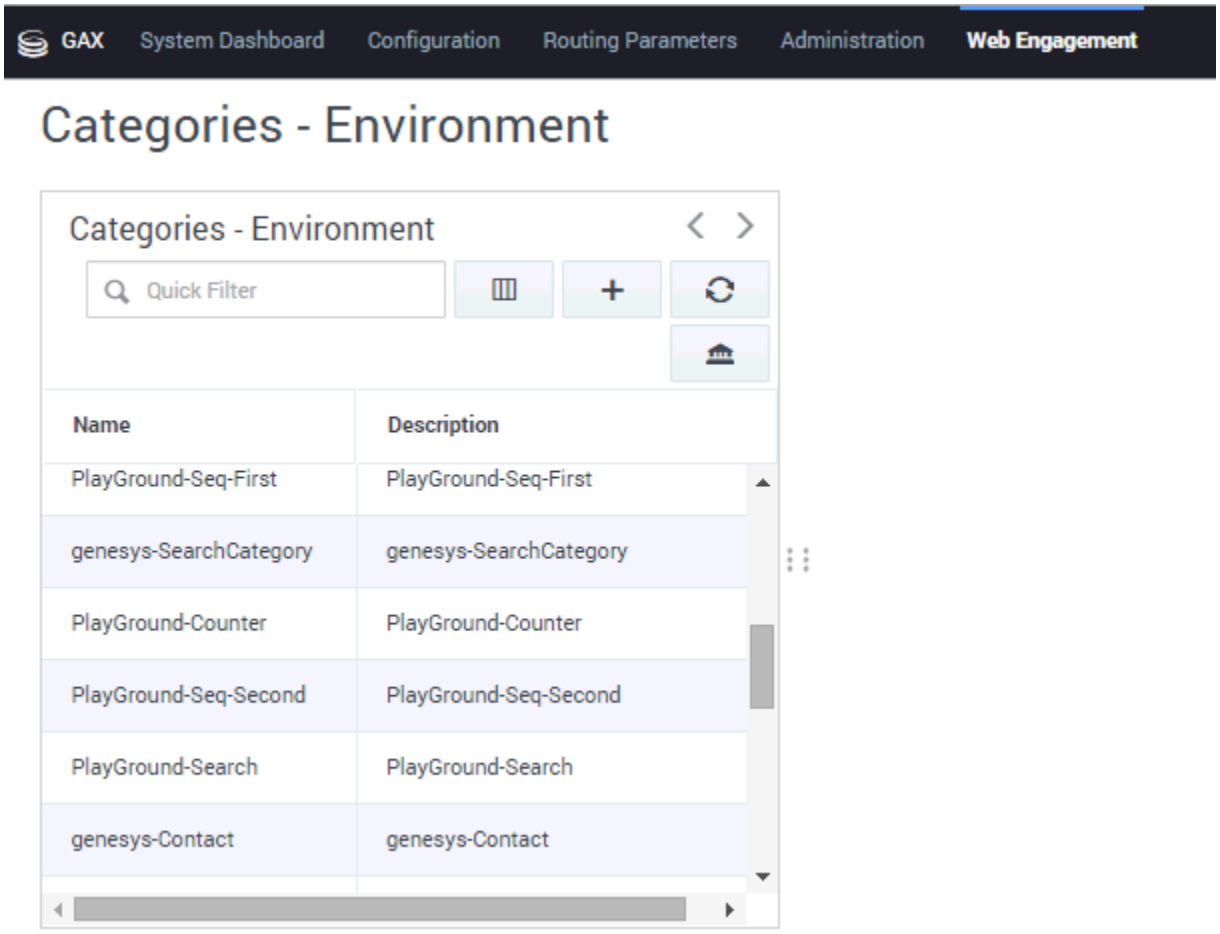
You can use categories with both types of event.

About the GAX Plugin

You can add and remove categories for Web Engagement through the Category interface in the Genesys Administrator Extension plugin. You create these categories during the Application Development process if you use the Simple Engagement Model when you Create Business Information.

Each category is compliant with the category definition and includes tags to define business

information related to your website. To access the `Categories` interface, open Genesys Administrator Extension and navigate to `Web Engagement > Categories`.

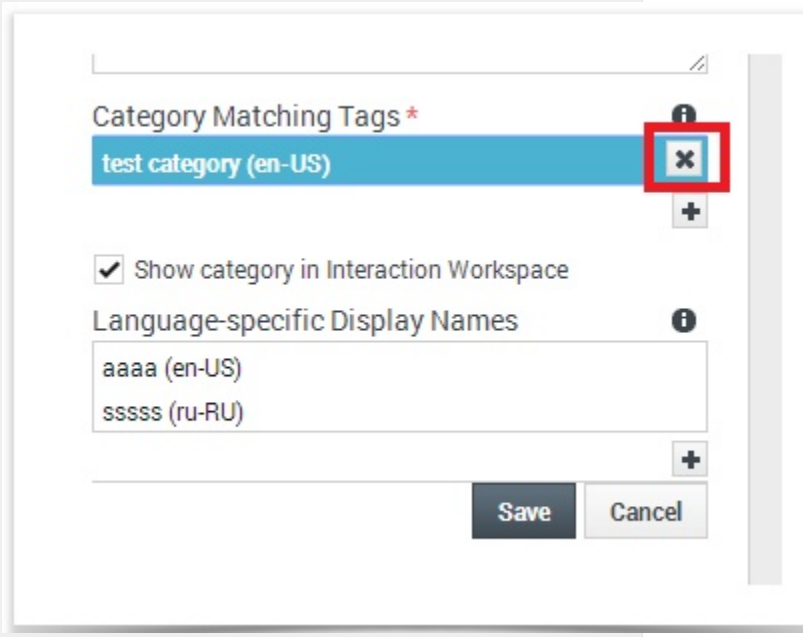
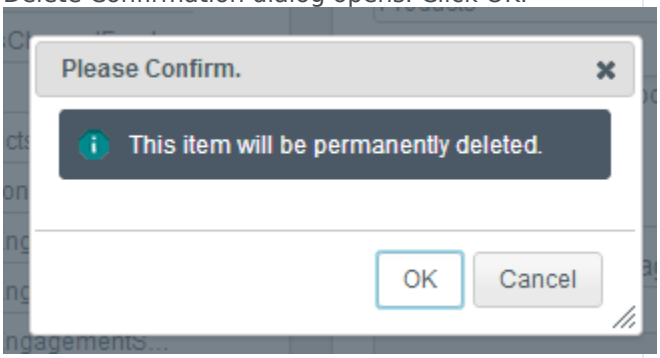


A list of Categories

Features

The Categories interface includes the following features:

Feature	Usage
Create categories.	See Creating a Category for instructions.
Create matching tags.	See Creating Category Matching Tags for instructions.
Delete matching tags.	Select the tag in the Category Matching Tag section and click X.

Feature	Usage
	 Click Delete.
Delete categories.	Select the category in the list and click Delete. The Delete Confirmation dialog opens. Click OK.  Delete Confirmation.

Important

You can also find the categories in Configuration Manager, but you should not edit or delete them through that interface because it can cause synchronization issues with the Categories interface in GAX.

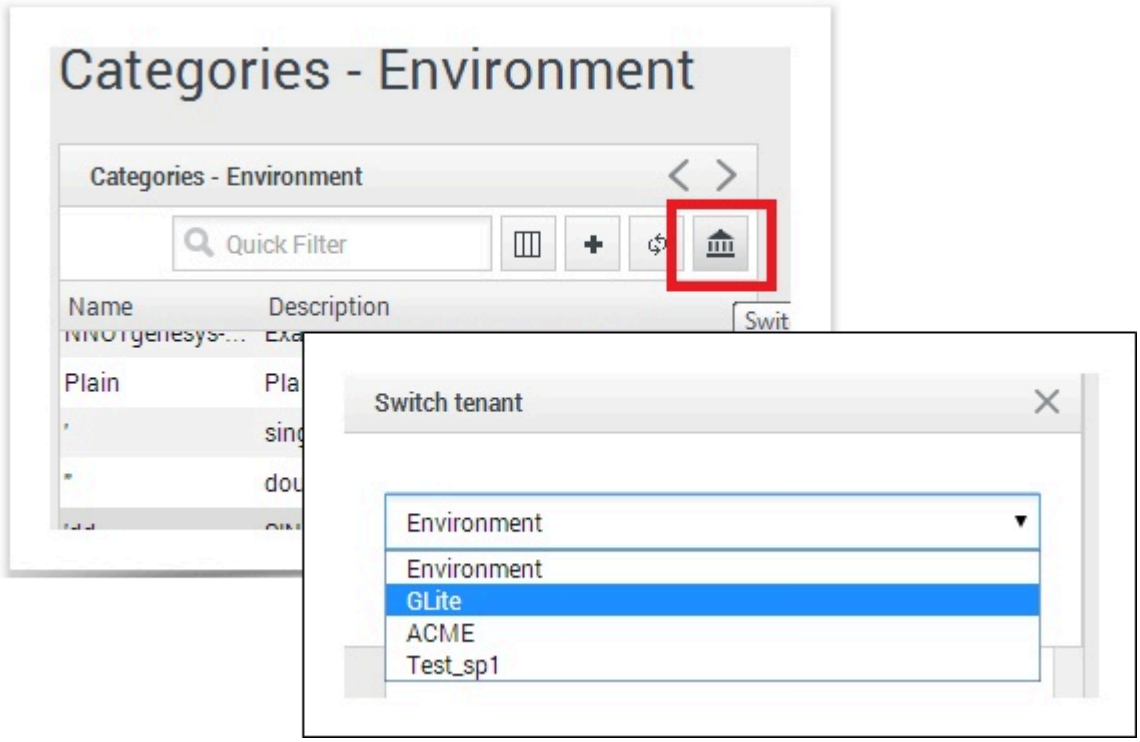
Creating a Category

Prerequisites

- Your environment includes Genesys Administrator Extension. See Genesys environment prerequisites for compliant versions.
- You installed the Web Engagement Plugin for Genesys Administrator Extension.

Start

1. In Genesys Administrator Extension, navigate to `Web Engagement > Categories`. The Categories interface opens.
2. Click `Switch Tenant`, select the tenant where you deployed Genesys Web Engagement, and click OK.



Click the Switch tenant.

3. Click + to add a new Category. The New panel opens.
4. Enter a `Category Name`. For instance, `pfs-login`.
5. Optionally, you can enter a `Category Description`.
6. Enable `Show category in Interaction Workspace` to display this category in Interaction Workspace if an agent opens interactions that are related to it.
7. Click Save. The `Products` category is added to the list.

End

Creating Category Matching Tags

Each category should have at least one Category Matching Tag, which contains an expression to search in the URLs and titles submitted with the events of the browser. For instance, you could identify an inbound voice–related page, such as http://www.genesys.com/products/genesys-inbound-voice/overview.aspx by using plain text, such as `genesys-inbound-voice`, or a regular expression like `'Inbound Voice'`.

Prerequisites

- You completed Creating a Category.

Start

1. In Genesys Administrator Extension, navigate to `Web Engagement > Categories` and select a category. The `<category name>` panel opens.
2. In the `Category Matching Tags` section, click +. The `New` panel opens.
3. Fill in the form to create a tag. Consult the table below for more information about the form fields.

Field	Description
Name	The display name for your tag. For example, `Inbound Voice`.
Type	The type of expression to search. There are three options: • `Regular Expression` — A regular expression search. • `Plain Text` — A substring search. This is the default. • `Google Like Expression` — Selecting this option opens a new window where you can enter an expression using Google search operators. When you click `Generate to REGEX`, it converts the expression to a regular expression and populates the `Expression` field.
Expression	The expression to search. This can be plain text or a regular expression.
Case-sensitive	Selecting this field makes the regular expression case-sensitive. It is not selected by default.
Language	Select the language for the tag. This allows you to make the search expression specific to the localization of the browser.

4. Click Save. The tag is added to the list of `Category Matching Tags`.

5. If needed, you can also define display names for the category that are language specific. In the `Language-specific DisplayNames` section click +. The `New` panel opens.
6. Enter a `Name`.
7. Select a `Language`.
8. Click Save. The language-specific display name is added to the list on the `<category name>` panel.
9. Click Save on the `<category name>` panel.

End

Regular Expressions in Tags

You can create tags that use regular expressions to search for matches by selecting "Regular Expression" from the Type list. A regular expression is a sequence of elements, either a word or expression inside quotes. Each search element can be preceded by a '-' to exclude that element. A wildcard symbol '*' can be used inside or outside of the quotes. If you prefer, you can select "Google Like Expression" for the Type, which converts anything you enter in the "Expression" field to a regular expression. If your expression is incorrect, your expression is not converted.

Search Request Patterns (Google Like Expression)

The following table describes the patterns in search requests.

Search Options	Description
Search for all exact words in any order. *search query*	The result must include all the words. These words can be substrings attached to other words—for example, [Web-search query1].
Search for an exact word or phrase. "*search query*"	Use quotes to search for an exact word or set of words in a specific order without normal improvements such as spelling corrections and synonyms. This option is handy when searching for song lyrics or a line from literature—for example, ["imagine all the people"].
Exclude a word. *-query*	Add a dash (-) before a word to exclude all results that include that word. This is especially useful for synonyms like Jaguar the car brand and jaguar the animal. For example, [jaguar speed -car].
Include "fill in the blank". *query *query*	Use an asterisk (*) within a query as a placeholder for any terms. Use with quotation marks to find variations of that exact phrase or to remember words in the middle of a phrase. For example, ["a * saved is a * earned"].

Managing Business Events

Overview

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) 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 name="">
        </event>
        -->

        <!-- This is template for your search event -->
        <!--
        <event id="Search" name="Search">
                <trigger name="SearchTrigger" element="" action="click" url="" count="1" />
                <val name="searchString"                value="" />
        </event>
        -->

        <event id="Timeout-30" name="Timeout-30">
                <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 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 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 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 or the browser registers keyboard input for any element on the page.

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.

Creating Business Events by Customizing the DSL File

You can edit the **apps*Your application name*\resources\dsl\domain-model.xml** and add a list of events, with specific conditions, related to your web pages' content.

> **Important**
>
> Genesys recommends that you use the InTools application to help you modify your DSL.

The default **domain-model.xml** file provides the Timeout-30 event and a template for Search events to help you get started with your customizations. The following sections show how you can customize these events for use on your website.

Using the SearchEvent Template

By default, the **domain-model.xml** file contains commented code that you can implement to trigger a business event when a visitor tries to search for something on your website. Complete the following steps to customize the SearchEvent for your website.

Start

1. Remove the comment characters that wrap around the event: <!– and -->. The event should look like the following:

```
<event name="Search">
    <trigger name="SearchTrigger" element="" action="click" url="" count="1" />
    <val name="searchString"       value="" />
</event>
```

2. Set the **element** attribute to the jQuery selector that triggers a search. For example, we have an input (`id="search"`) with a submit button (`id="search-submit"`).

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

3. Set the **value** attribute to the script to retrieve the search string. For example, our input id of "search".

```
<event name="Search">
    <trigger name="SearchTrigger" element="#search-submit" action="click" url=""
count="1" />
    <val name="searchString" value="$(#search).val()" />
</event>
```

Now the search event is triggered when a visitor clicks the **search-submit** button.

End

Using the Timeout Events

By default, the **domain-model.xml** file contains the timeout-30 event.

```
<event id="Timeout-30" name="Timeout-30">
    <trigger name="TimeoutTrigger" element="" action="timer:30000" type="timeout" url=""
count="1" />
</event>
```

By default, this event is triggered 30 seconds after the tracking script is initialized on the page.

Creating Business Events by Using the Monitoring Agent API

You can also use the Monitoring JS API, which allows you to submit events and data from the HTML source code.

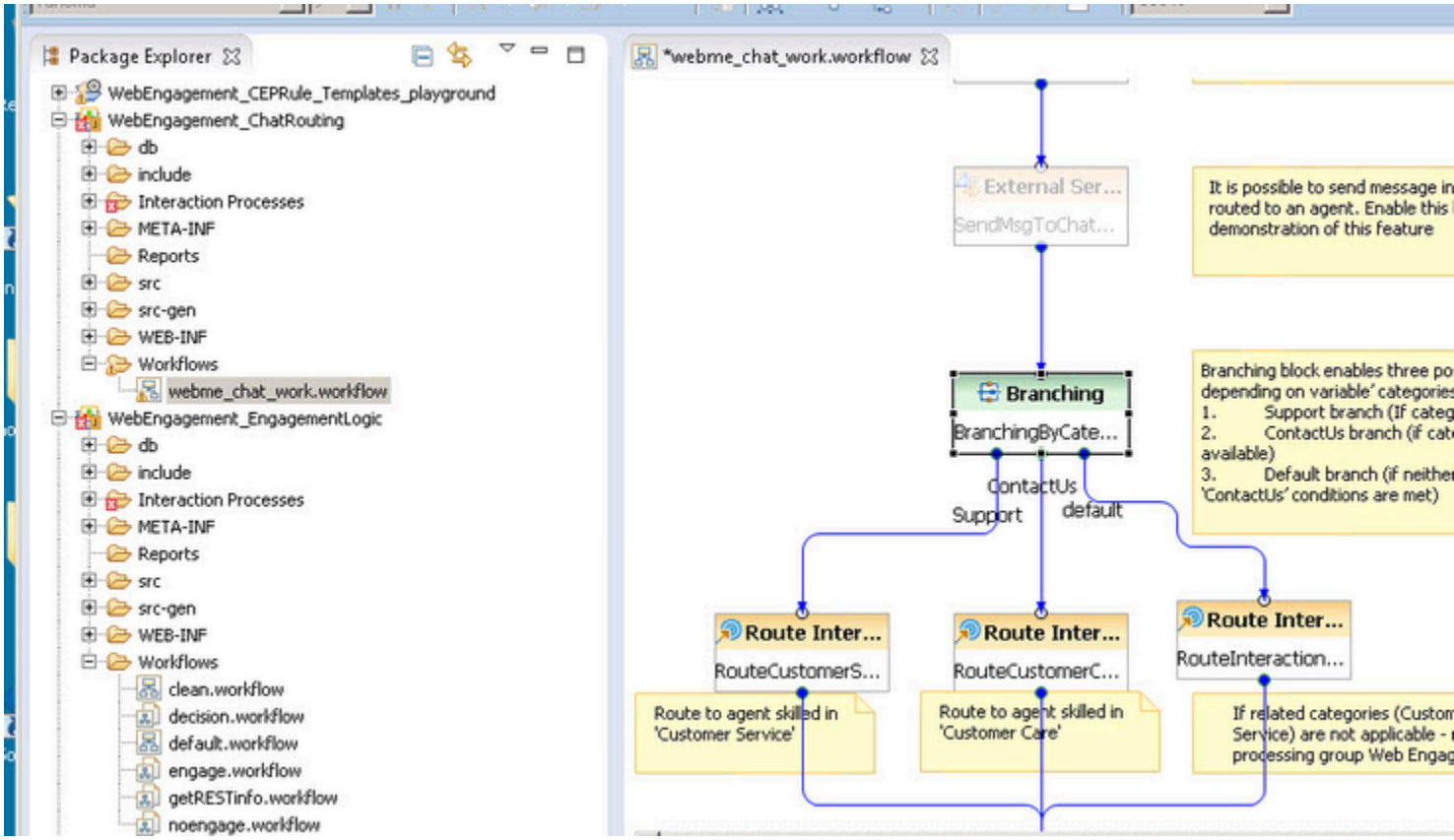
In this case, you can use the `_gt.push()` method which allows you to decide when events should be submitted and which data they generate, directly from your web pages. See Monitoring JS API Reference for further details.

You should also consider using the API when you have more complex logic that can't be handled by DSL alone. For an example, see How To — Enable a trigger after another trigger.

Managing Routing 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.

You can customize the routing strategies to help meet your specific business needs:

- Customizing the Engagement Strategy
- Customizing the Chat Routing Strategy

Managing Rules

Overview

Rules are mandatory for managing actionable events generated from the System and Business event flows submitted by the Browser Tier. To add rules, you must create a package and then a set of rules. For details about rules, refer to the Genesys Rules System documentation.

Multi-Package Domain Oriented Rules

As of version 8.1.2, Genesys Web Engagement supports multi-package domain oriented rules. You can map your rules package to a particular domain by reversing the domain zone in the name of the rules package. For example, the `blog.genesys.com` domain would have a rules package called `com.genesys.blog`.

You can have multiple rules packages on the same server at the same time. New rules packages (with a different package name) that are deployed do not rewrite the current rules, but are instead added to the current rules set. When the existing rules package is deployed, it rewrites selected package rules in the current rules set.

This domain mapping is applied hierarchically - the "root" domain is processed by the "root" package and the sub-domain is process by the sub-package and all parent packaged (including "root").

For example, your website contains the following sub-domains:

- genesys.com
- blog.genesys.com
- communication.genesys.com
- personal.communication.genesys.com

And you have the following rules packages:

- com.genesys
- com.genesys.blog
- com.genesys.communication
- com.genesys.communication.personal

The rules packages are processed as follows:

Domain	com.genesys	com.genesys.blog	com.genesys.communication	com.genesys.communication.p
genesys.com	+	-	-	-

Domain	com.genesys	com.genesys.blog	com.genesys.communication	com.genesys.communication.p
blog.genesys.com	+	+	-	-
communication.genesys.com	+	-	+	-
personal.communication.genesys.com	+	-	+	+

> **Important**
>
> This feature is turned off by default. You can turn on domain separation rule execution on the specified Web Engagement server by setting the domainSeparation option to `true`.

Creating a Rules Package

Complete the steps below to create the rules package associated with your Web Engagement application. This procedure is an example of how to create a rules package. For further information about creating rules, refer to the Genesys Rules System Deployment Guide.

Prerequisites

- Your environment includes Genesys Rules Authoring Tool. See Genesys environment prerequisites for compliant versions.
- Roles are configured to enable your user to create rules.

Start

1. Open the Genesys Rules Authoring Tool and navigate to `Environment > Solution > New Rule Package`.
2. In the `General` tab:
   - Enter a `Package Name`. For example, `myproject.rules.products`.
   - Enter a `Business Name`. For example, `Products`.
   - Select `web_engagement` for `Package Type`. `WebEngagement_CEPRule_Templates` appears in the `Template` table.
   - Optionally, you can enter a `Description`.
3. Select `WebEngagement_CEPRule_Templates` in the `Template` table.

Create a new rules package

4. Click Save.

End

Creating Rules in the Rules Package

Prerequisites

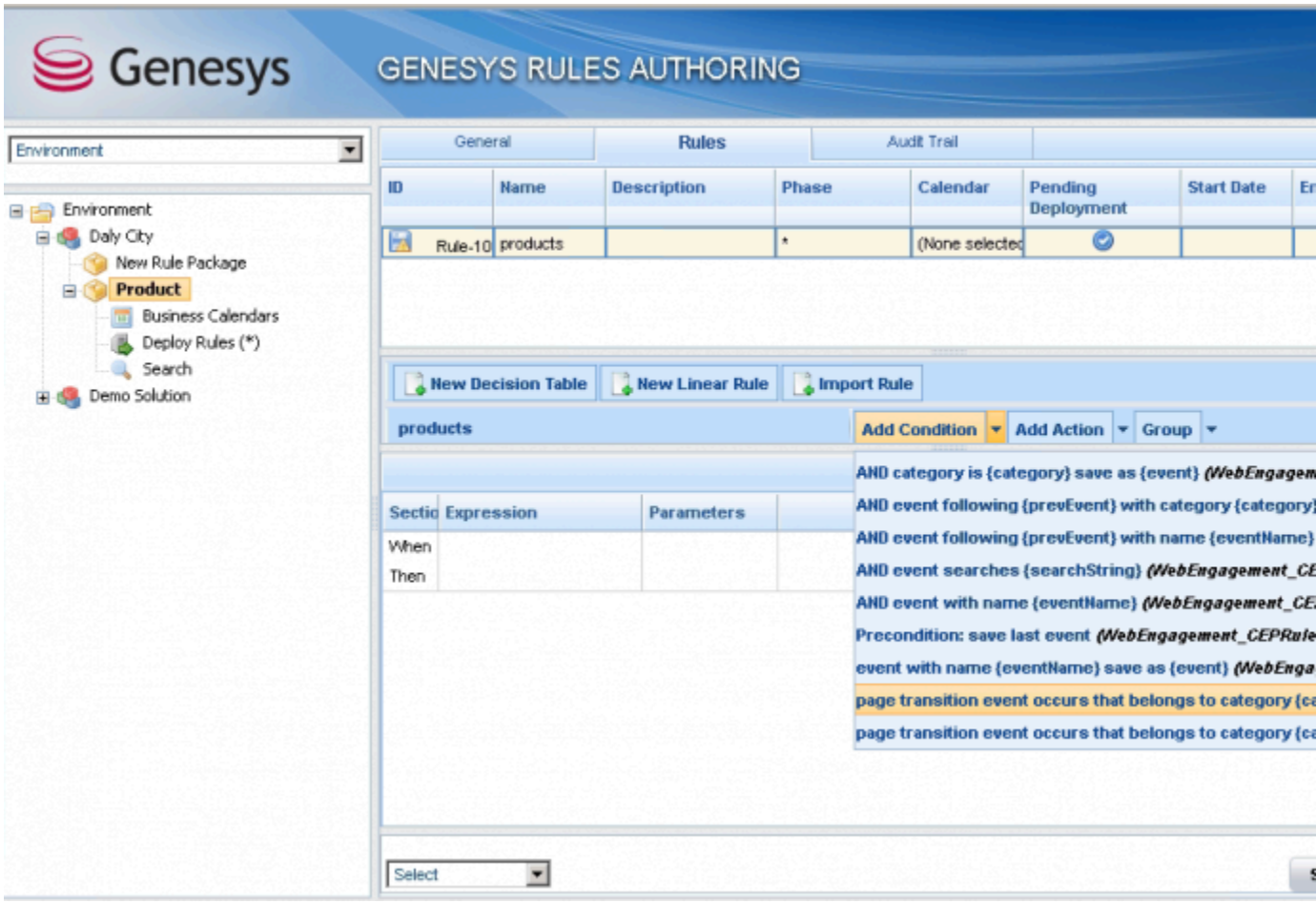
- Creating a Rules Package

Start

1. In Genesys Rules Authoring Tool, select the rules package you created in the previous procedure.
2. Select the `Rules` tab.
3. Click `New Linear Rule`. This creates a new rule in the `Rules` table.
4. Select the created rule:

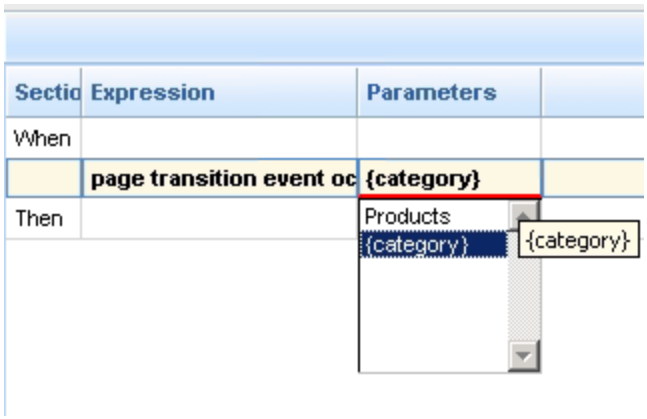
- Enter a Name. For example, Products.
- Enter a Phase. The list of rule phases can be modified by changing the values of the Phases enumeration in the CEP Rules Template. The default value is *.

5. Click Add Condition:
   - Scroll down to select a condition. For example, page transition event occurs that belongs to category, which launches the actionable event any time that a user enters or leaves a page on your website.



Select your rule's condition

   - Select a category in Parameters. For example, Products. The Parameters list displays the categories that you previously created.



Set the condition's parameters

6. Click `Add Action` and select an action in the list. For example, `generate actionable event`.
7. Click `Save...`.
   You can create as many rules as you need in your rules package.

End

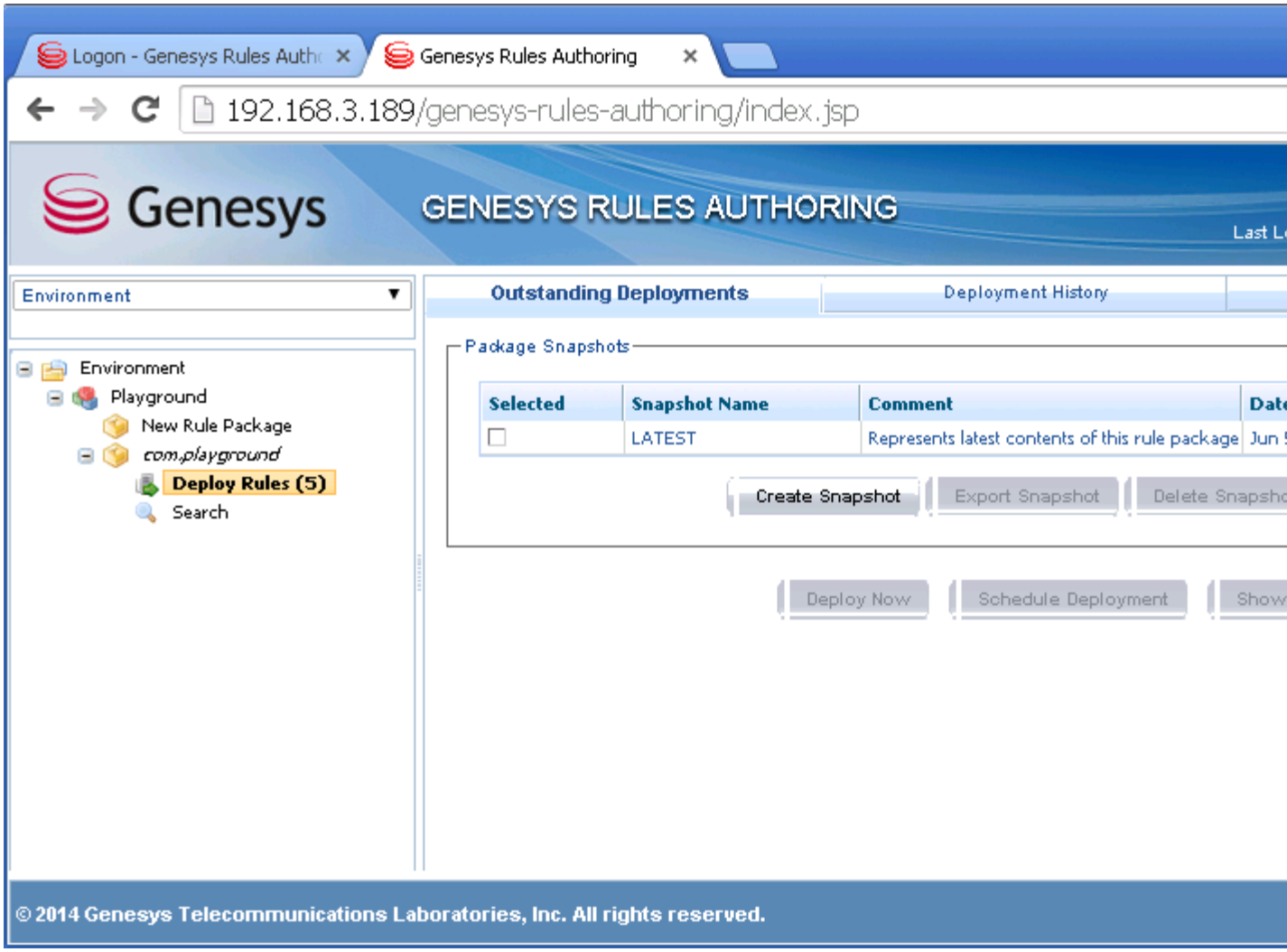
Deploying the Rules Package

Prerequisites

- You started the Web Engagement servers.

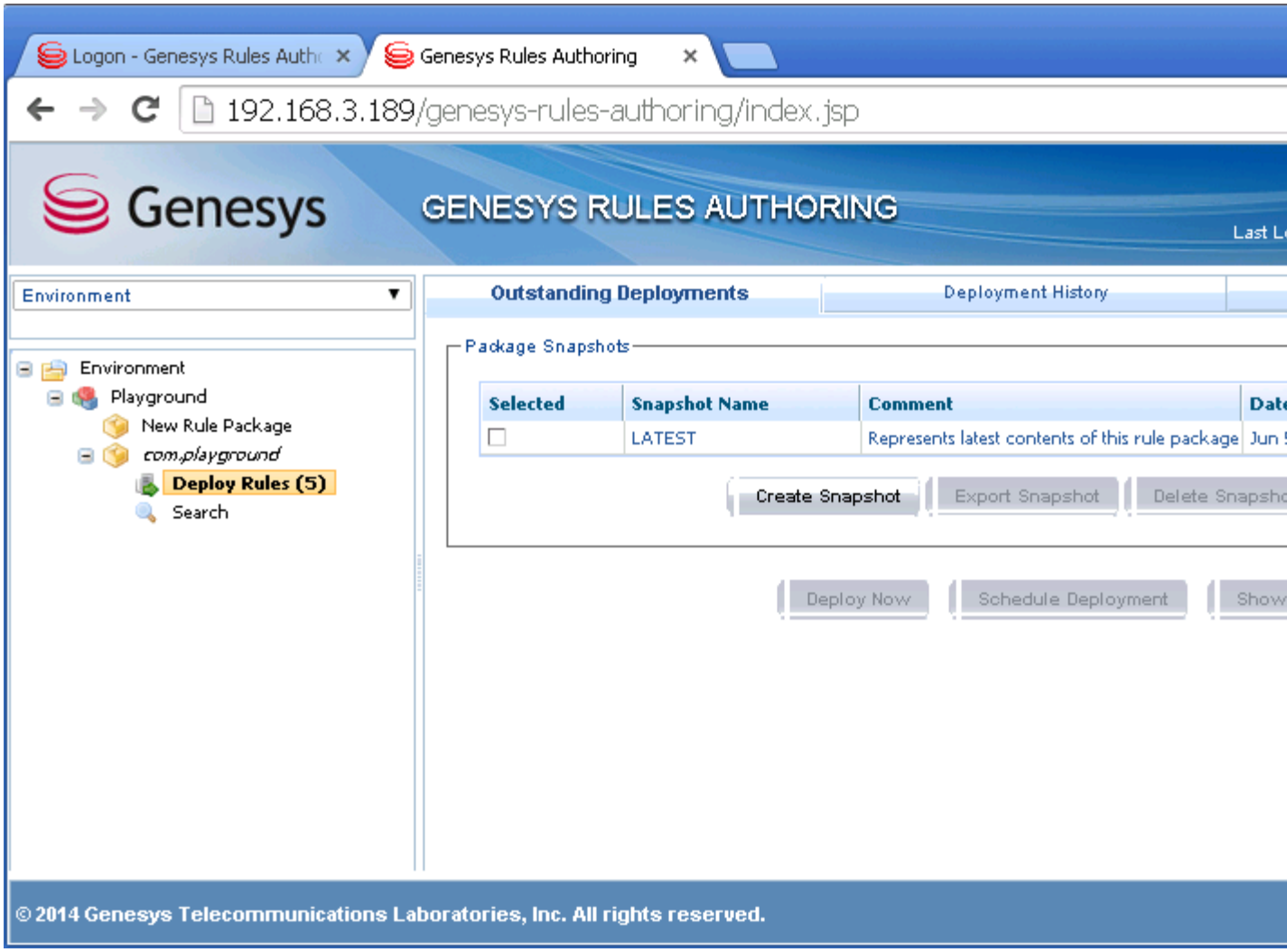
Start

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



Deploy the rules package

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**.



Deploy the rules package

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

End

Using InTools

Overview

InTools (Instrumentation Tools) is a Chrome/Chromium extension you can use to create, validate, and test DSL.

Installation

You can install and use InTools with the Chromium Browser or the Chrome Browser. Genesys recommends that you use the Chrome Browser because it has the largest selection of Developers Tools.

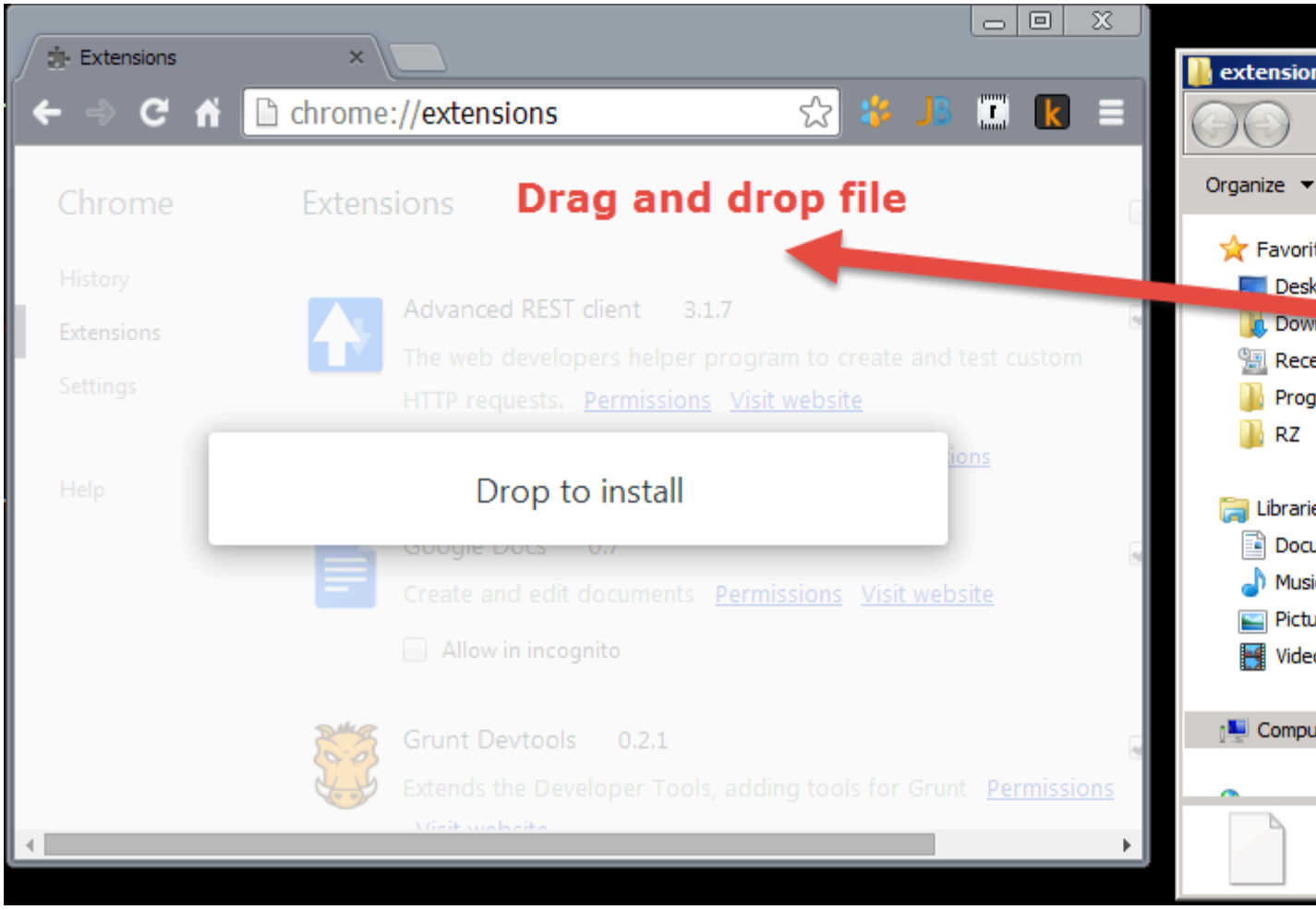
Drag-and-drop installation

This installation process is the same for Chromium (Windows, Mac, Linux) and Chrome (Mac, Linux).

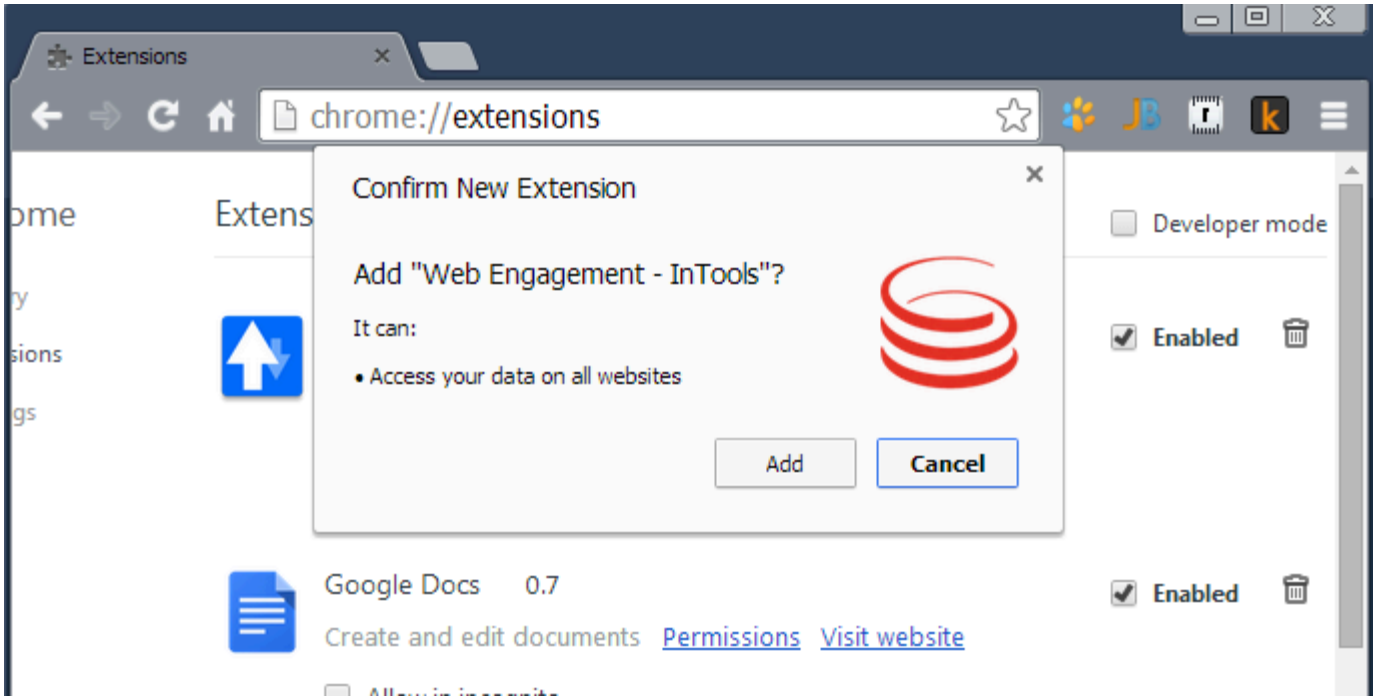
> **Important**
>
> Chrome in Windows has some restrictions about installing extensions that are not from the Chrome Store. If your browser blocks the installation of the extension or displays an error message, install it as unpacked extension in developer mode. See Installation as an unpacked extension for instructions.

Start

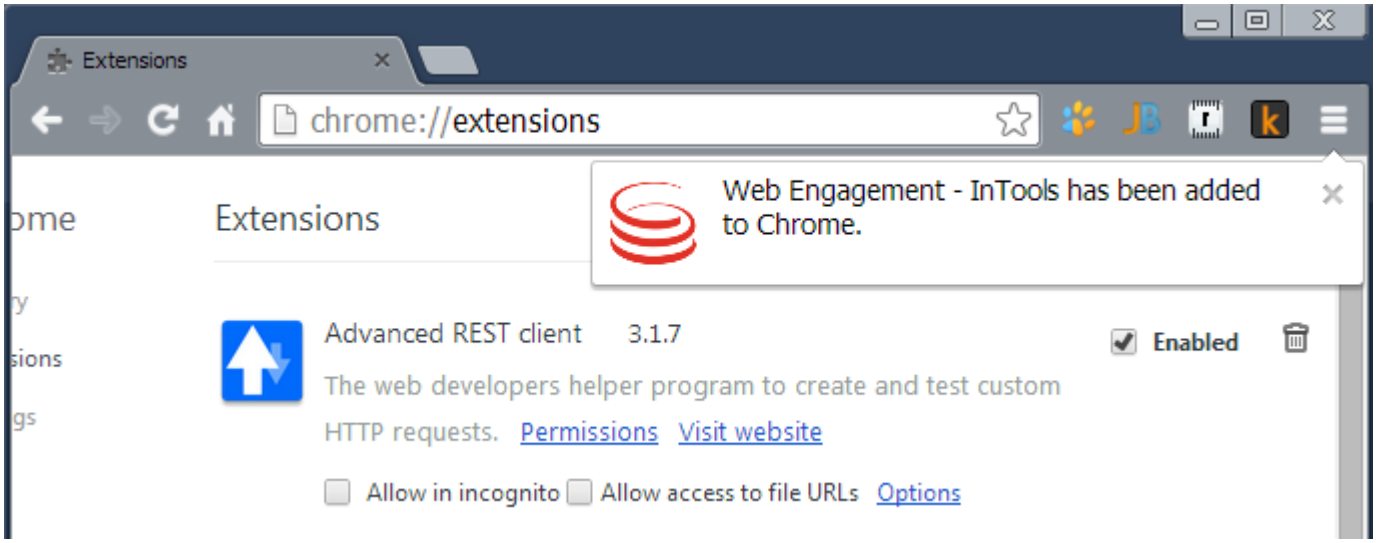
1. Run the Chrome browser.
2. Navigate to the **chrome://extensions** page.
3. Open a window in your OS and navigate to the ***GWE_installation*\tools\intools\extension** directory.
4. Drag and drop the **intools.crx** file to the browser.



5. In the **Confirm New Extension** dialog, click **Add**.



The InTools extension is added to the extension list and enabled:



.

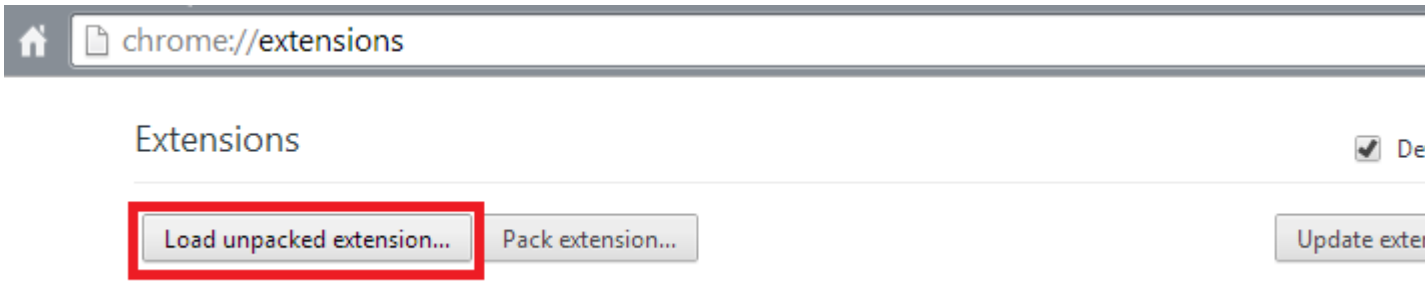
End

Installation as an unpacked extension

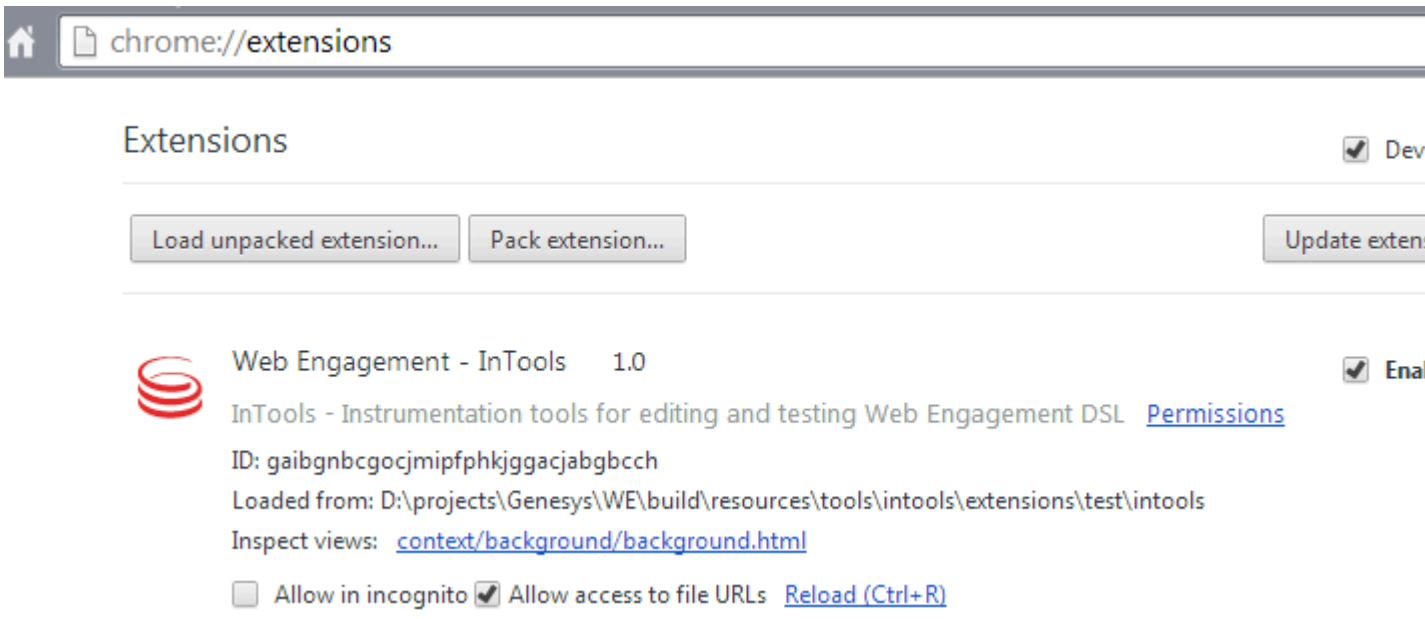
1. Run the Chrome browser.

2. Navigate to the **chrome://extensions** page.
3. Enable Developer Mode.

4. Open a window in your OS and navigate to the ***GWE_installation*\tools\intools\extension** directory.
5. Unpack **intools.crx** by using a zip archiver.
6. Click **Load unpacked extension...** and choose the directory with InTools that you unpacked in the previous step.

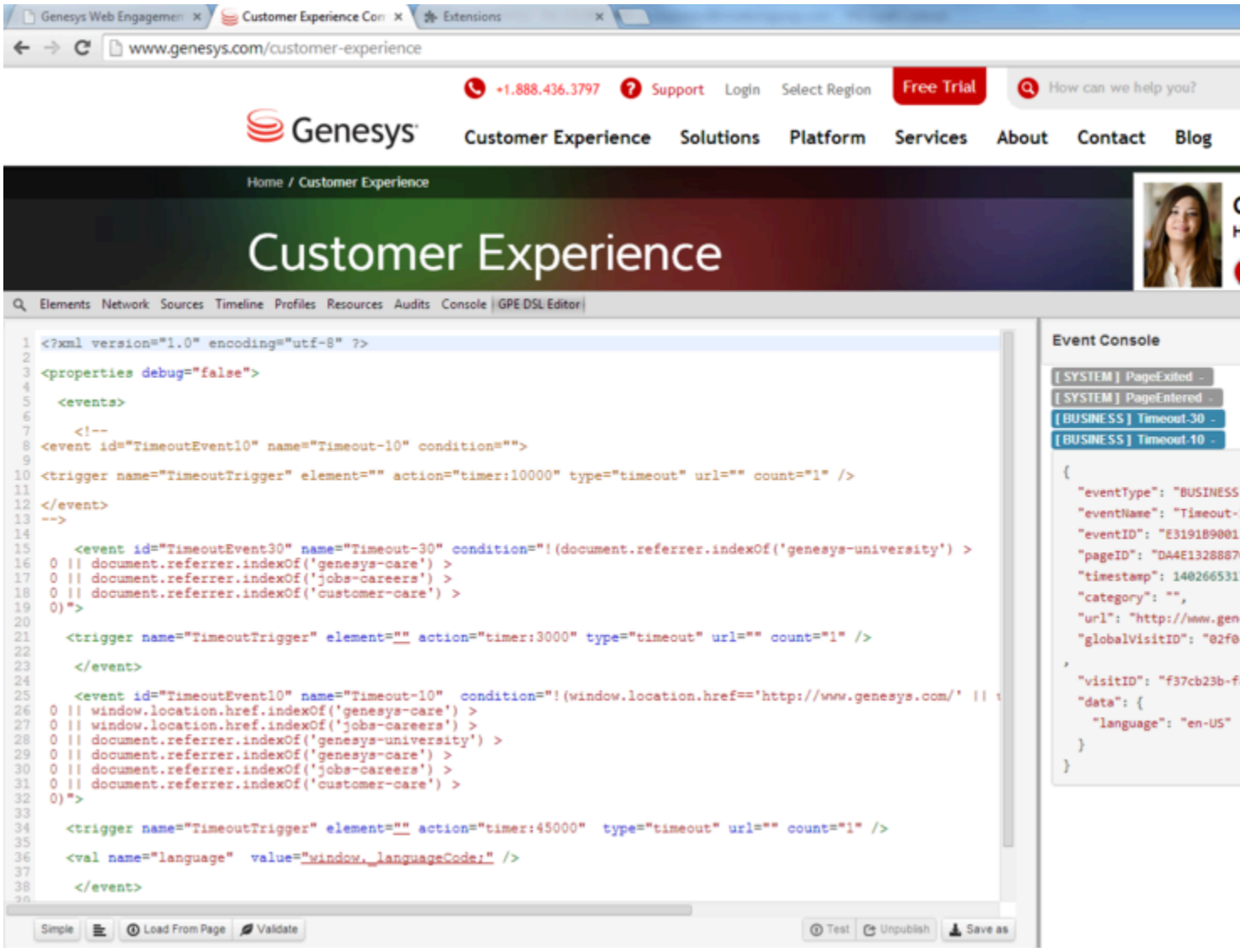


The InTools extension is added to the extension list and enabled:



GWE DSL Editor

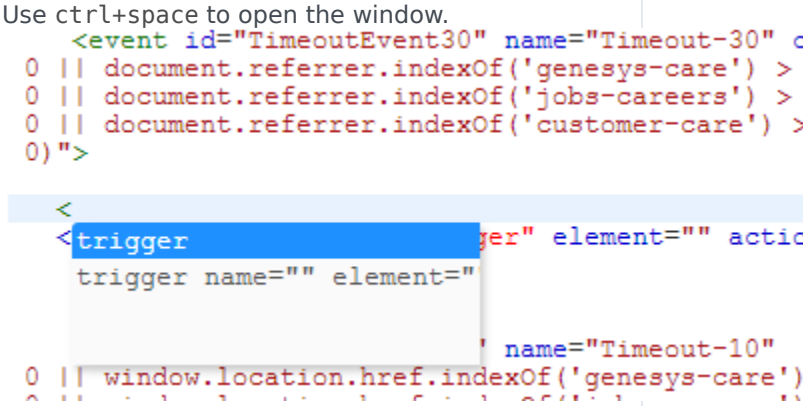
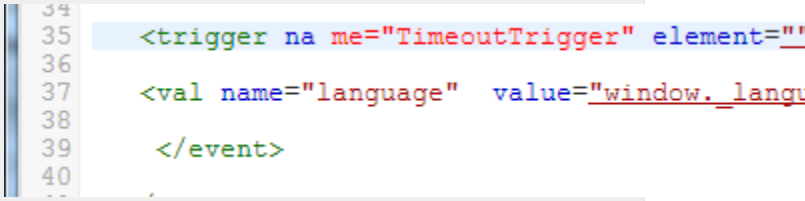
The DSL Editor allows you to create, edit, and test Genesys Web Engagement DSL.

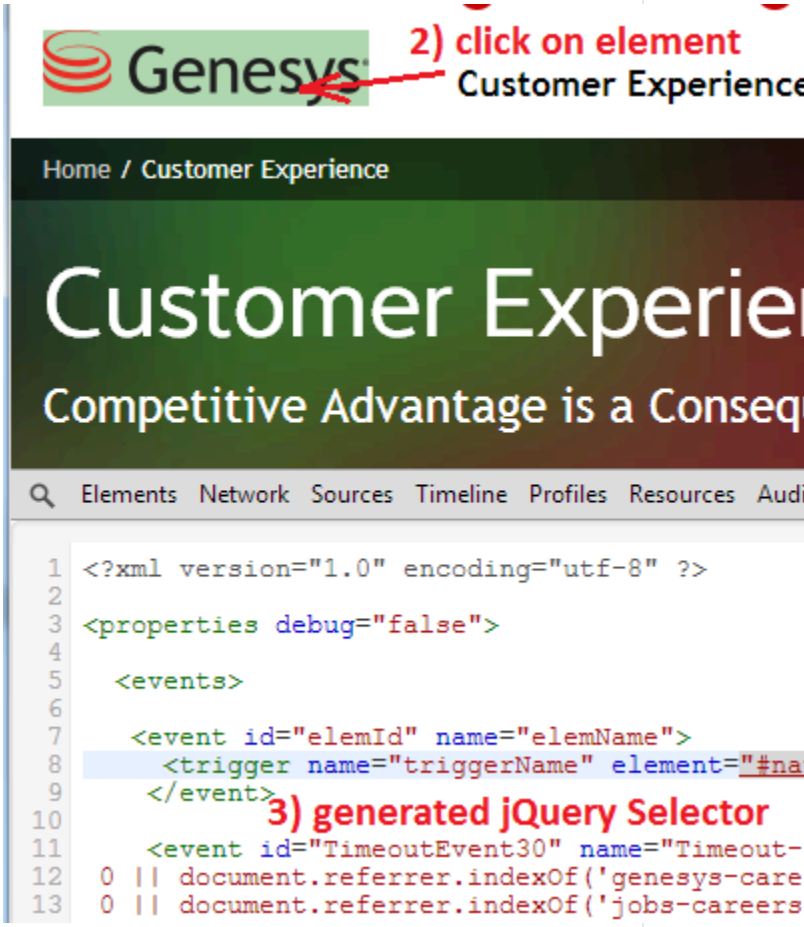


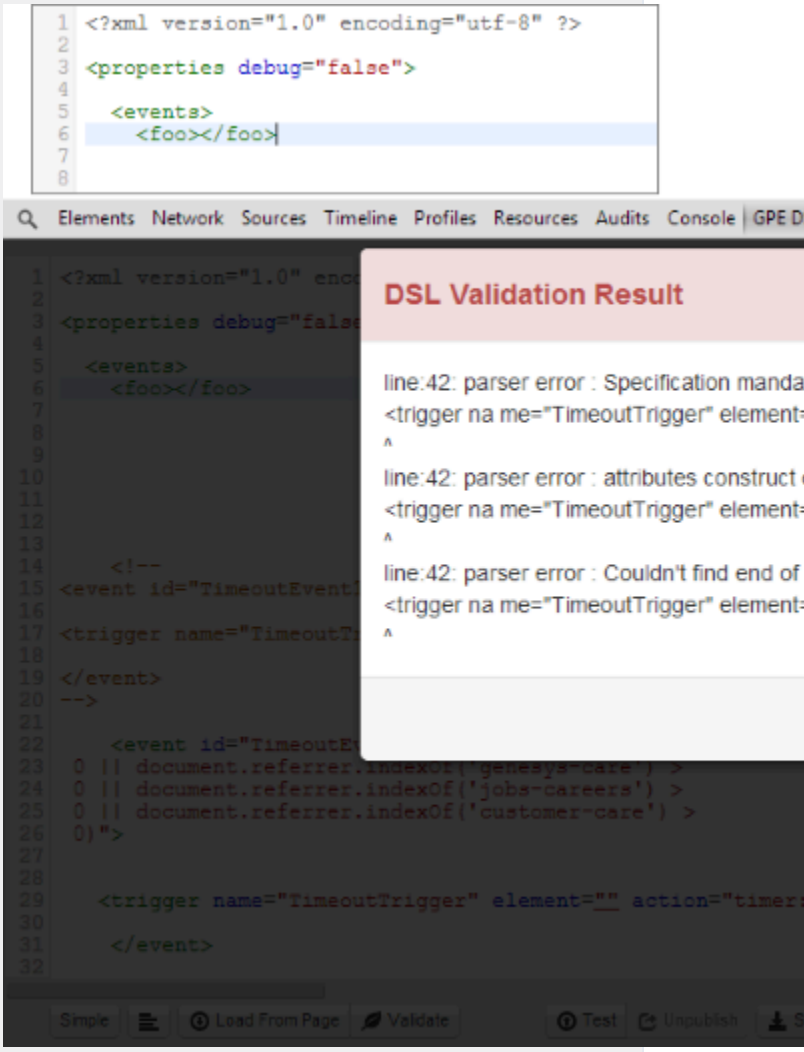
The DSL Editor

Advanced Editor

The Advanced Editor is a text editor for Genesys Web Engagement DSL that provides the following features:

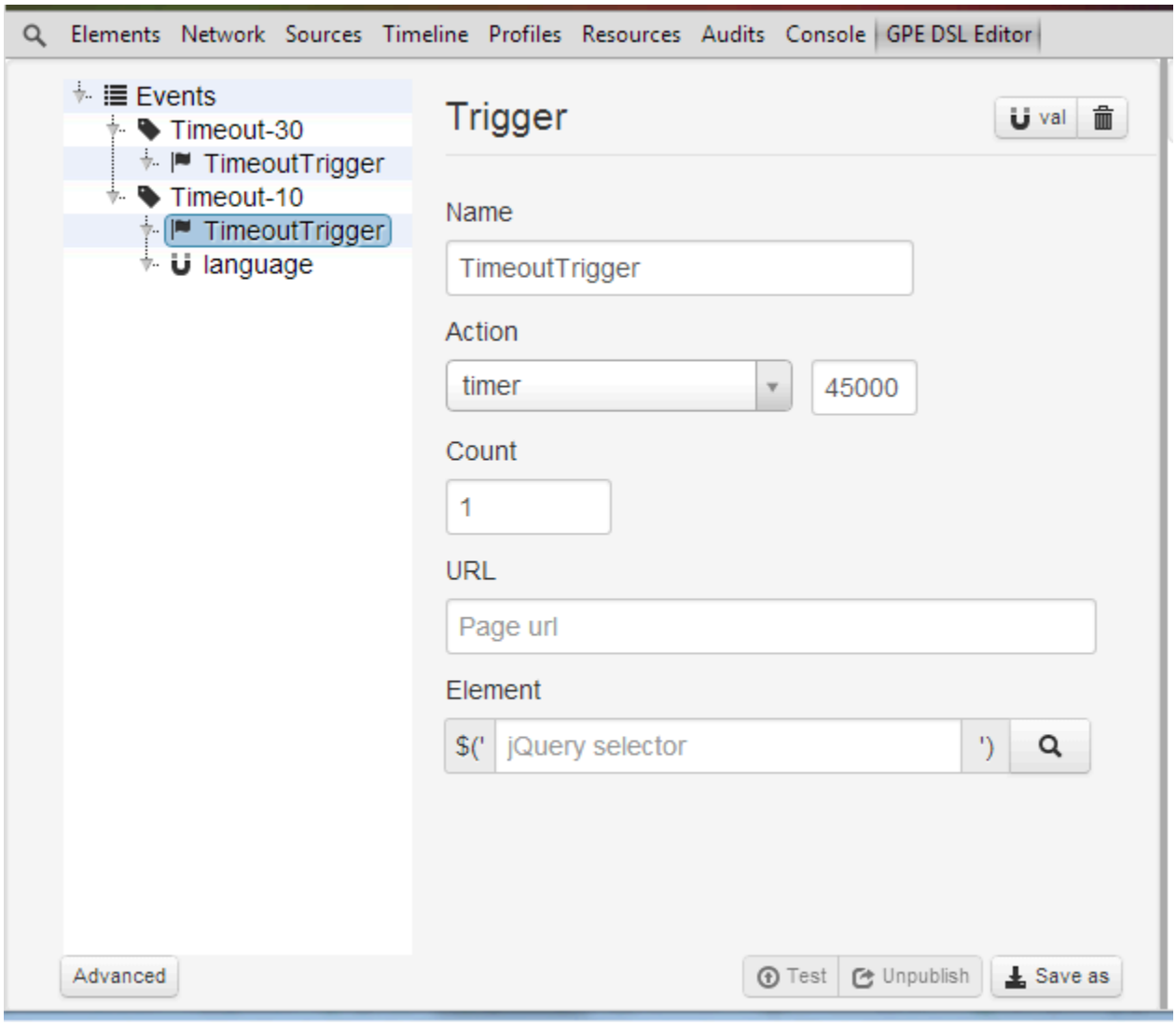
Feature	Usage
Syntax highlighting. The editor highlights reserved words to improve readability.	Automatic.
Code complete. The editor provides a code complete menu so you can see options to auto-complete your code.	Use `ctrl+space` to open the window.  The code complete menu.
XML syntax validation.	Automatic.  Syntax validation

Feature	Usage
jQuery selector highlighting. The tool can generate values for underscored attributes ("element" and "value") by clicking HTML elements on the page.	 Selector highlighting
DSL structure validation. The tool validates the names and positions of elements, including children.	Click Validate.

Feature	Usage
	 An example of DSL validation
Load the current DSL from a page. You can navigate to an already instrumented page and load DSL from it.	Click Load From Page.
Pretty formatting.	Click .
Test DSL. You can publish DSL changes to the current page	Click Test. The page is restarted and configured with the new DSL.
Unpublish DSL. You can removed the DSL you published.	Click Unpublish.
Save as. You can save your DSL changes to the file system.	Click Save as.

Simple Editor

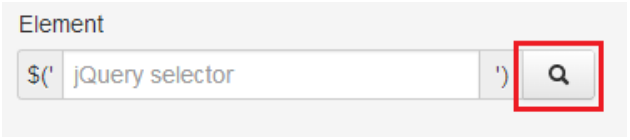
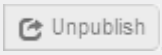
The Simple Editor is a wrap for the Advanced Editor that provides easy to use forms.



The Simple Editor

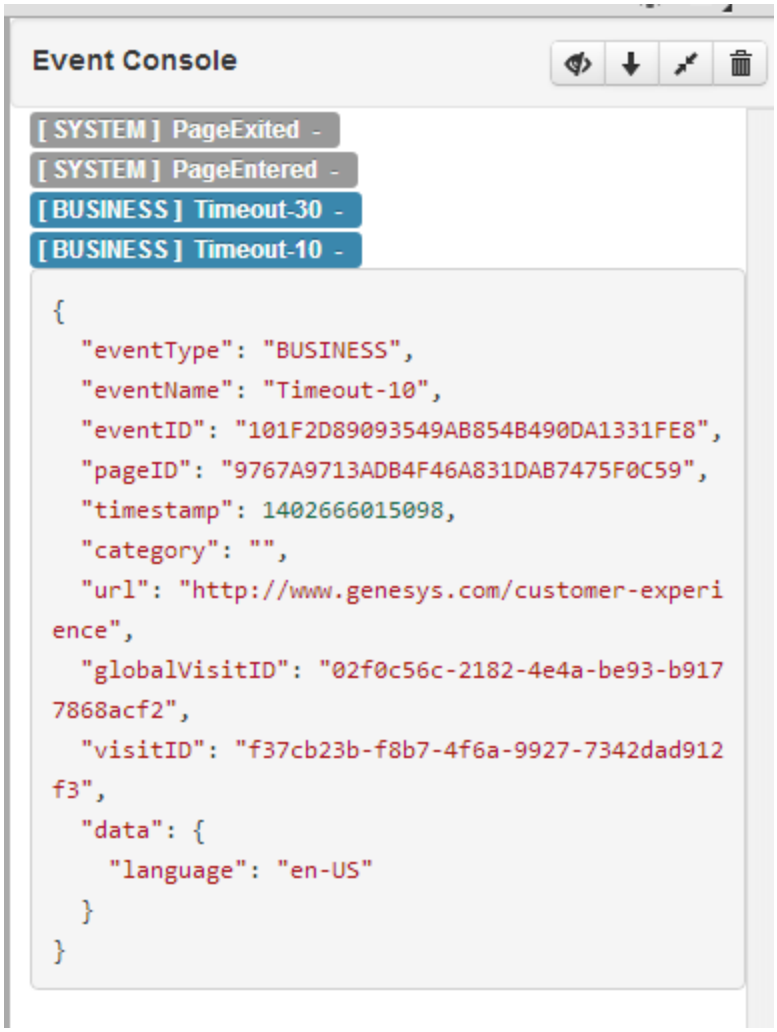
The editor includes the following features:

Feature	Usage
Navigation tree. The navigation tree allows you to quickly see events, triggers, and values.	Automatic.
jQuery selector highlighting. As in advanced mode, the tool can generate values for underscored attributes ("element" and "value").	Click the magnifying glass

Feature	Usage
	 .
Create or remove elements in the DSL tree.	Click 
Test DSL. You can publish DSL changes to the current page	Click  . The page is restarted and configured with the new DSL.
Unpublish DSL. You can removed the DSL you published.	Click  .
Save as. You can save your DSL changes to the file system.	Click  .

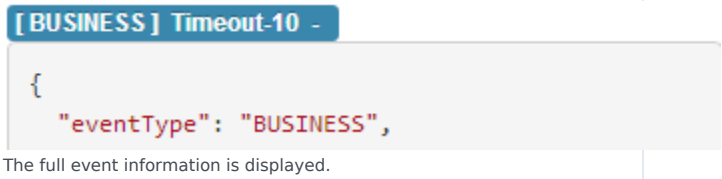
Event Console

The Event Console monitors the generated events (both system and business). The event body includes all the event data in JSON format. The same format is posted to the Web Engagement Server.



The Event Console

The Event Console includes the following features:

Feature	Usage
Generated events monitoring.	Automatic
Toggle the full information for an event.	Click the event name to hide/show the event information.  The full event information is displayed.  The full event information is hidden.

Feature	Usage
Hide all system events.	Click  .
Auto scroll to the last generated event.	Toggle  to enable.
Toggle all events.	Click  .