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.

User's Guide

Genesys Web Engagement 8.1.2

1/23/2022

Table of Contents

Genesys Web Engagement User's Guide

Welcome to the *Genesys Web Engagement 8.1.2 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.

Tools

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

Business Information

Learn how to manage business information.

Rules

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

Reporting

Find information to help you import and use GWE reporting templates.

Managing CEP Rule Templates

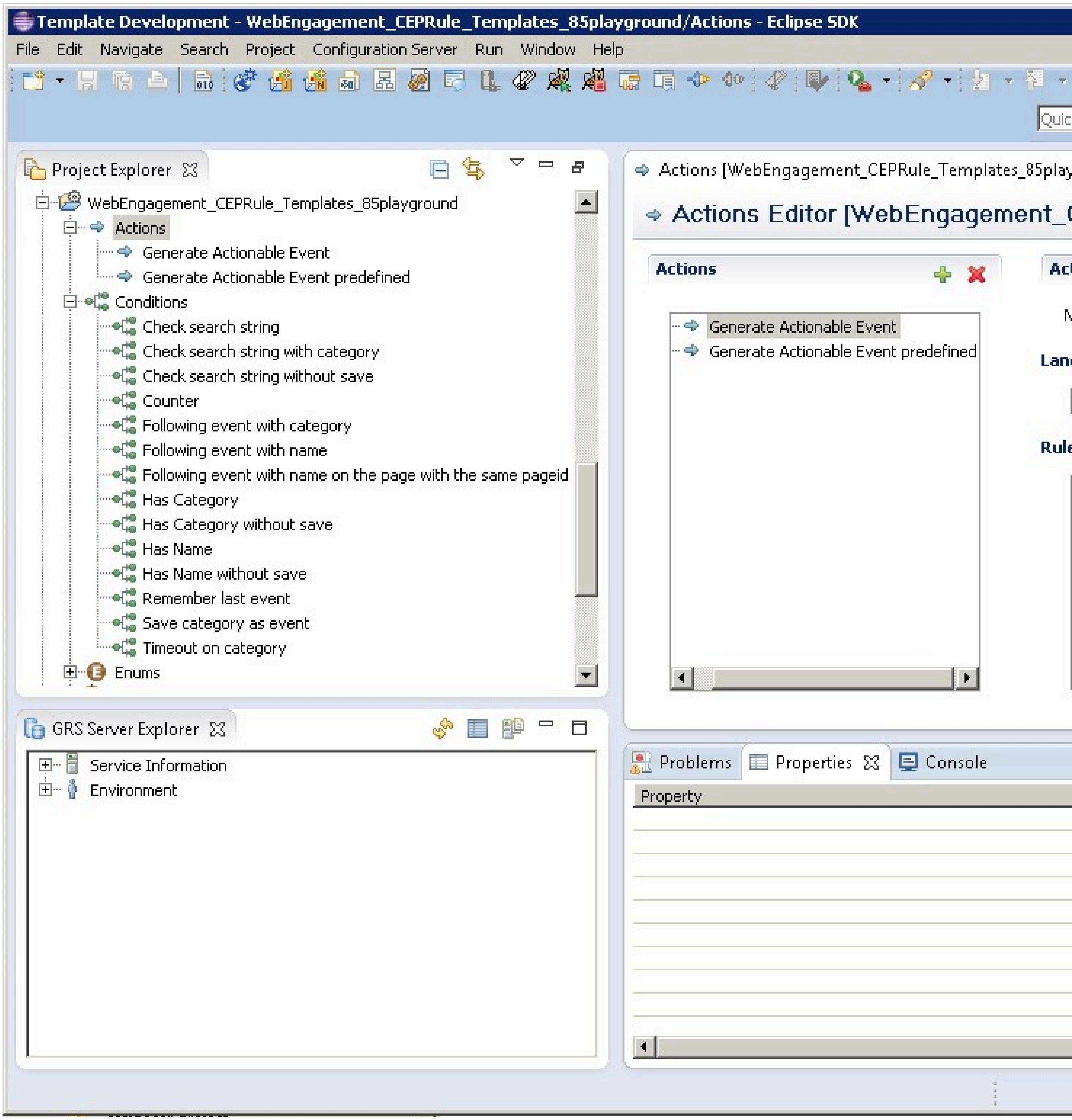
Overview

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 System (GRS) 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. GRS implements the CEP (Complex Event Processing) template for GWE. 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 rule templates are created with your Web Engagement application — in release 8.1.2, there are two CEP Rule Template projects:

- **\apps\*application name*\_composer-project\WebEngagement_CEPRule_Templates_85** includes templates that feature reworked conditions for 8.1.2.
- **\apps\*application name*\_composer-project\WebEngagement_CEPRule_Templates** includes templates that are compatible with Web Engagement 8.1.1.


Template Development - WebEngagement_CEPRule_Templates_85playground/Actions - Eclipse SDK
File Edit Navigate Search Project Configuration Server Run Window Help
Quic
Project Explorer
WebEngagement_CEPRule_Templates_85playground
Actions
Generate Actionable Event
Generate Actionable Event predefined
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 page with the same pageid
Has Category
Has Category without save
Has Name
Has Name without save
Remember last event
Save category as event
Timeout on category
Enums
GRS Server Explorer
Service Information
Environment
Actions [WebEngagement_CEPRule_Templates_85play
Actions Editor [WebEngagement_
Actions
Ac
N
Lan
Rule
Generate Actionable Event
Generate Actionable Event predefined
Problems
Properties
Console
Property

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:

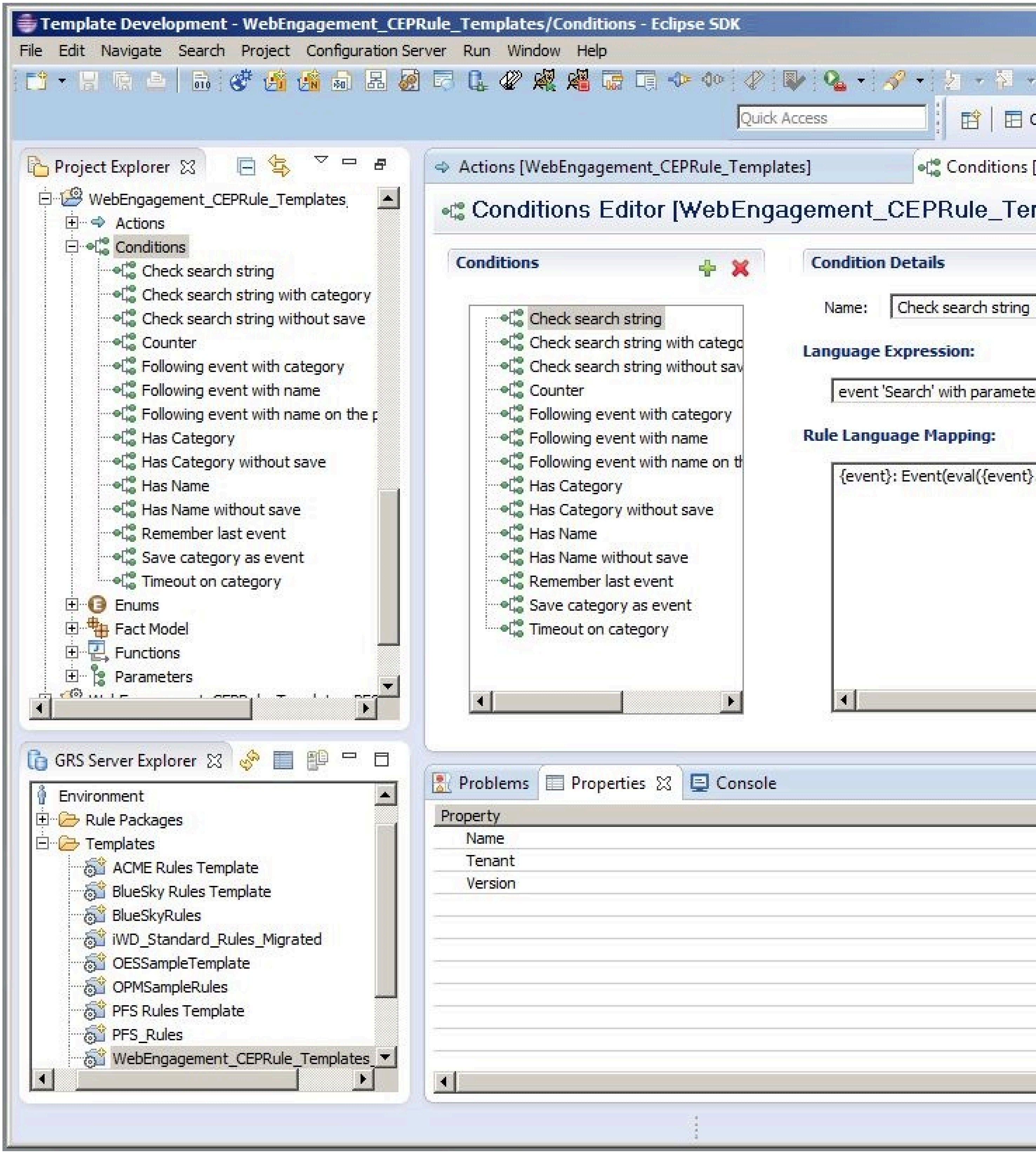
- Generate Actionable Event
- Generate Actionable Event Predefined

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**.



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

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

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

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

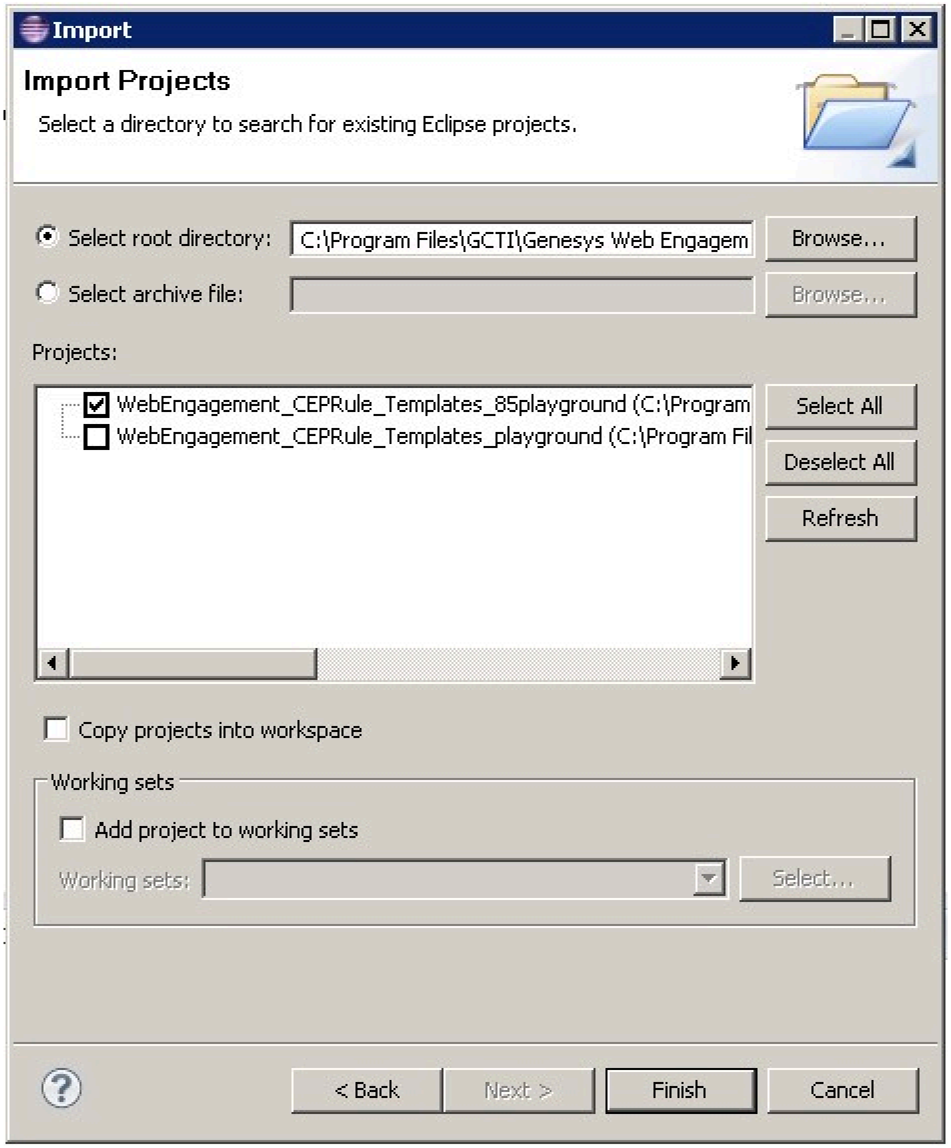
Complete this procedure to import the CEP rule templates in the Genesys Rules Development Tool. Even if you do not plan to customize the templates, your rule template must be published in the Rules System Repository before you try to create rules.

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. In release 8.1.2, Genesys Web Engagement includes two sets of rules templates: **_composer-project\WebEngagement_CEPRule_Templates** is compatible with 8.1.1, while **_composer-project\WebEngagement_CEPRule_Templates_85** is compatible with 8.1.2 and features reworked conditions. Select the rules template project to import:
   - Browse to the **\apps\*application name*\_composer-project** folder in the Genesys Web Engagement installation directory and select a project.
   - Click **OK**. **WebEngagement_CEPRule_Templates_85*application name*** is added to the **Projects** list.
   - Select the **WebEngagement_CEPRule_Templates_85*application name*** project.
   - Warning: Do **not** enable the option **Copy projects into workspace**.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\Program Files\GCTI\Genesys Web Engagem

○ Select archive file:

Projects:

☑ WebEngagement_CEPRule_Templates_85playground (C:\Program

□ WebEngagement_CEPRule_Templates_playground (C:\Program Fil

□ Copy projects into workspace

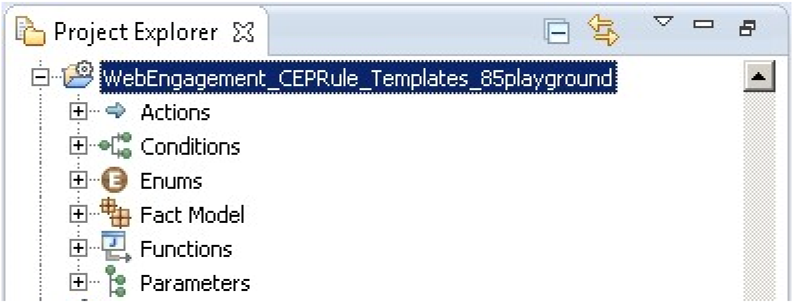
Working sets

□ Add project to working sets

Working sets:

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

- Click **Finish** to import the project. **WebEngagement_CEP_Rule_Templates_85*application name*** is added to the **Project Explorer**.



Project Explorer

WebEngagement_CEPRule_Templates_85playground
- Actions
- Conditions
- Enums
- Fact Model
- Functions
- Parameters

WebEngagement_CEP_Rule_Templates_85playground is added to the Project Explorer.

End

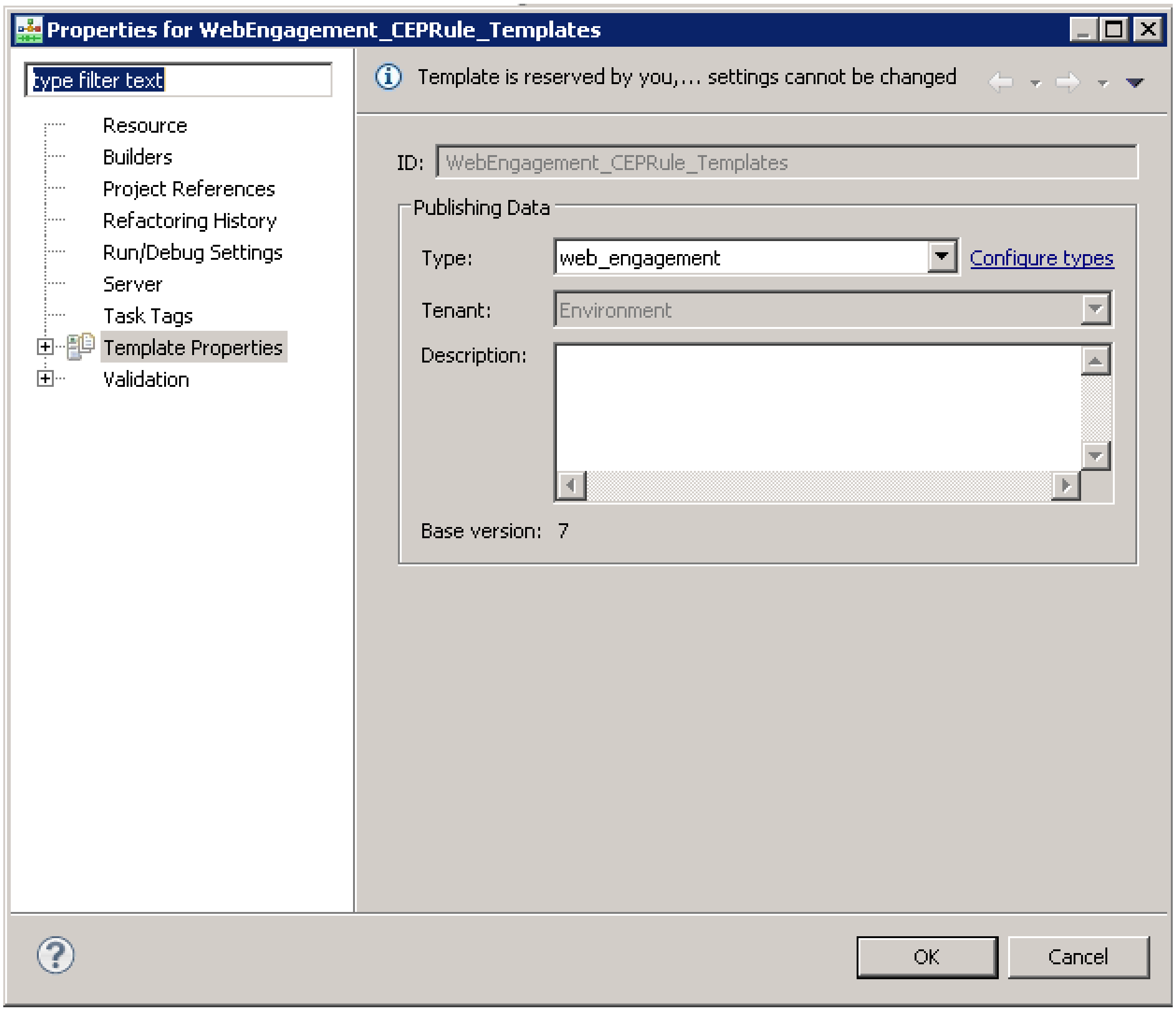
Configuring the CEP Rule Templates

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`.



Properties for WebEngagement_CEPRule_Templates

Template is reserved by you,... settings cannot be changed

ID: WebEngagement_CEPRule_Templates

Publishing Data

Type: web_engagement

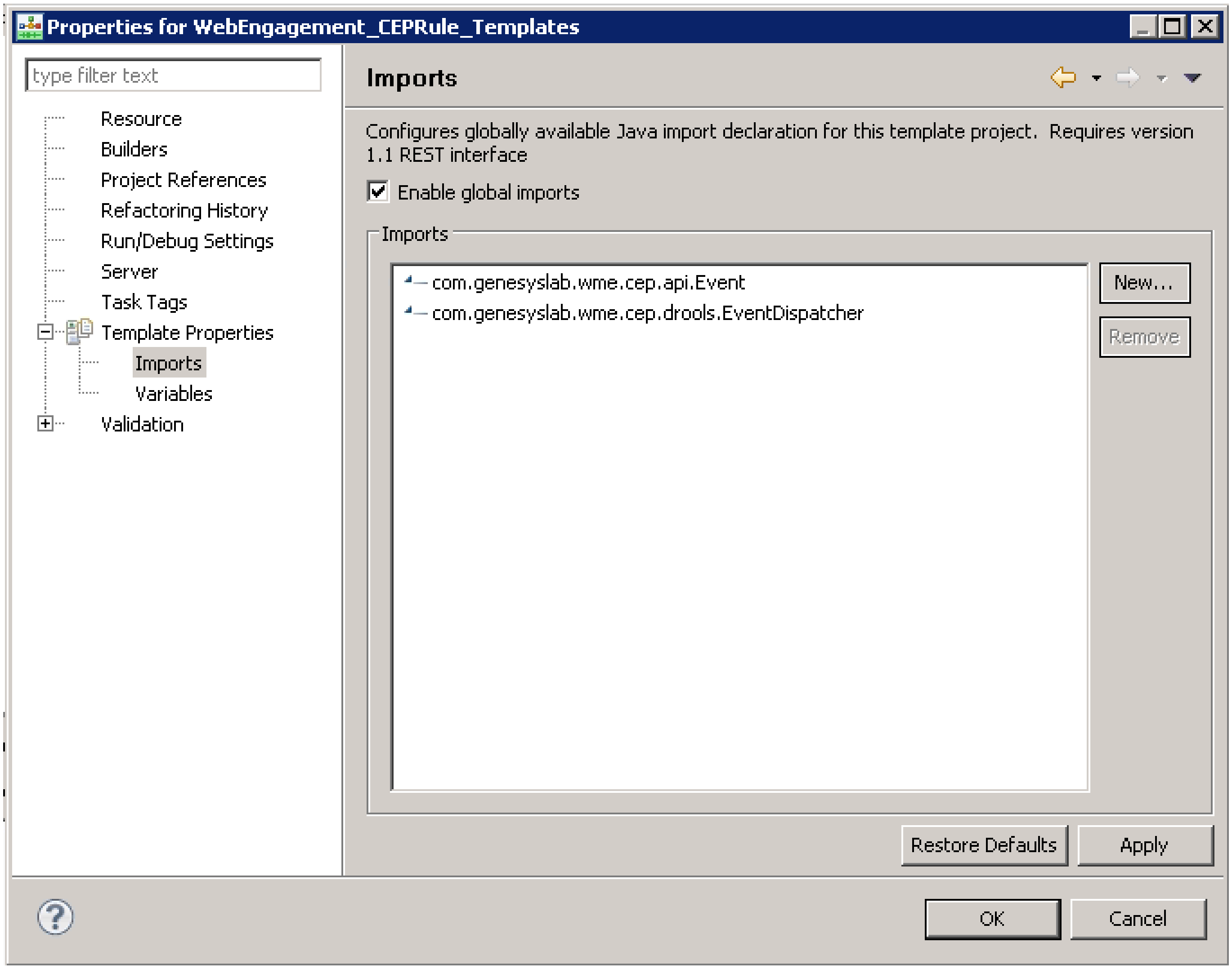
Tenant: Environment

Description:

Base version: 7

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

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



Configures globally available Java import declaration for this template project. Requires version 1.1 REST interface

☑ Enable global imports

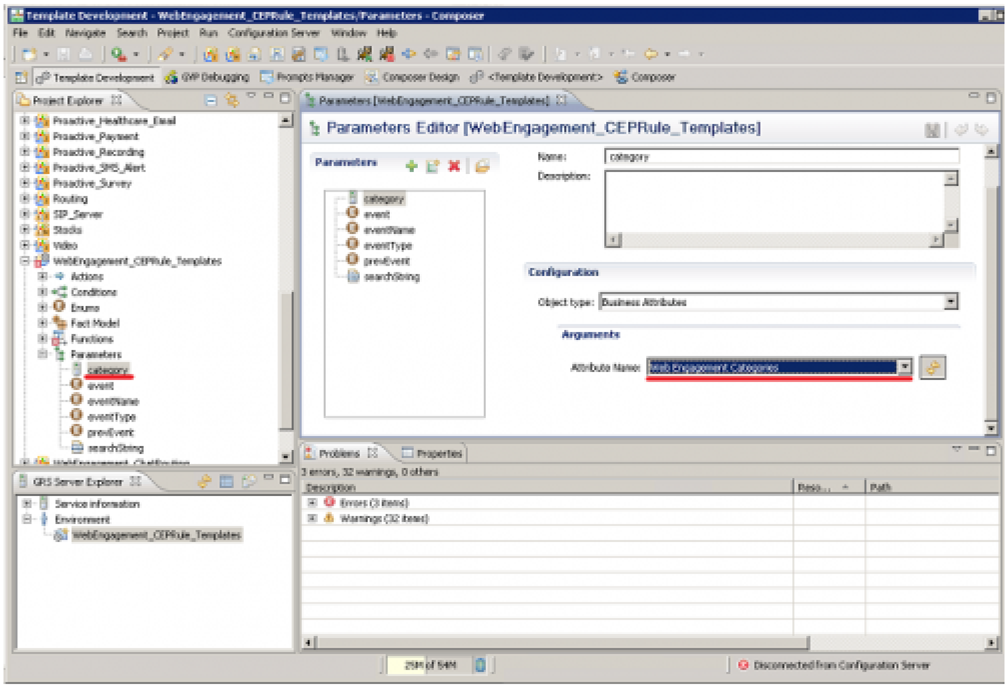
Imports

- com.genesyslab.wme.cep.api.Event
- com.genesyslab.wme.cep.drools.EventDispatcher

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`.



Name: category

Description:

Object type: Business Attributes

Attribute Name: Web Engagement Categories

3 errors, 32 warnings, 0 others

Errors (3 items)

Warnings (32 items)

Disconnected from Configuration Server

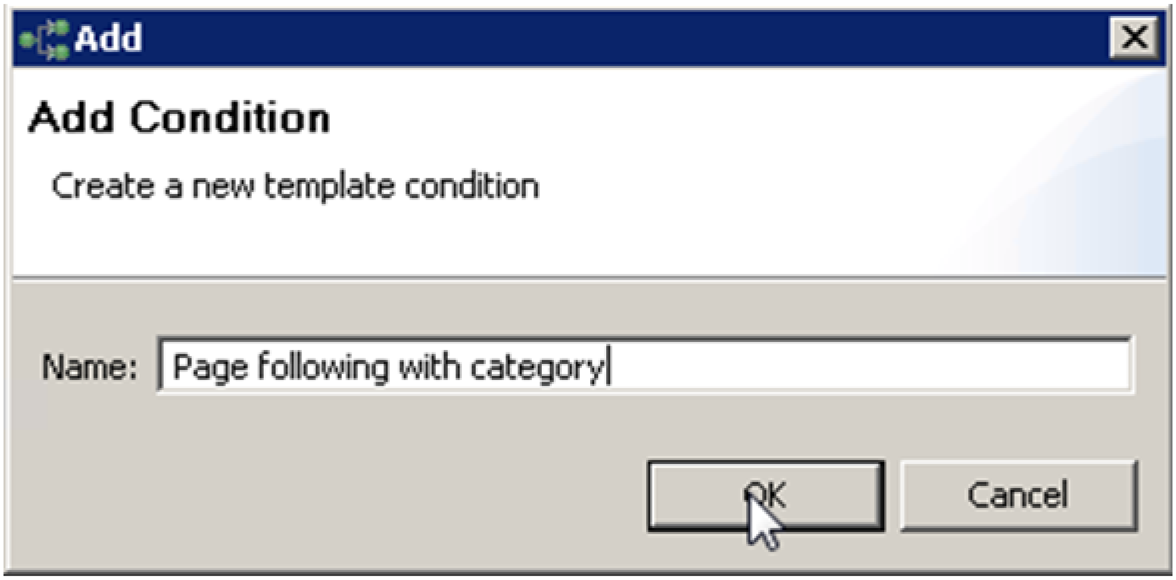
8. Click **Save**.

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

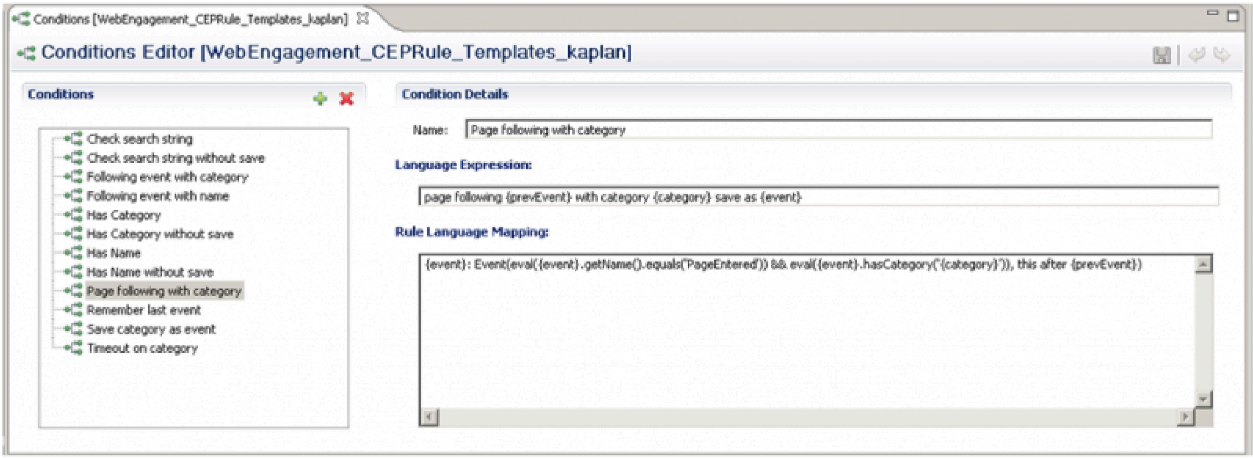
Add Condition

Create a new template condition

Name: Page following with category

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:



Conditions [WebEngagement_CEPRule_Templates_kaplan]

Conditions Editor [WebEngagement_CEPRule_Templates_kaplan]

Conditions

- Check search string
- Check search string without save
- Following event with category
- Following event with name
- Has Category
- Has Category without save
- Has Name
- Has Name without save
- Page following with category
- Remember last event
- Save category as event
- Timeout on category

Condition Details

Name: Page following with category

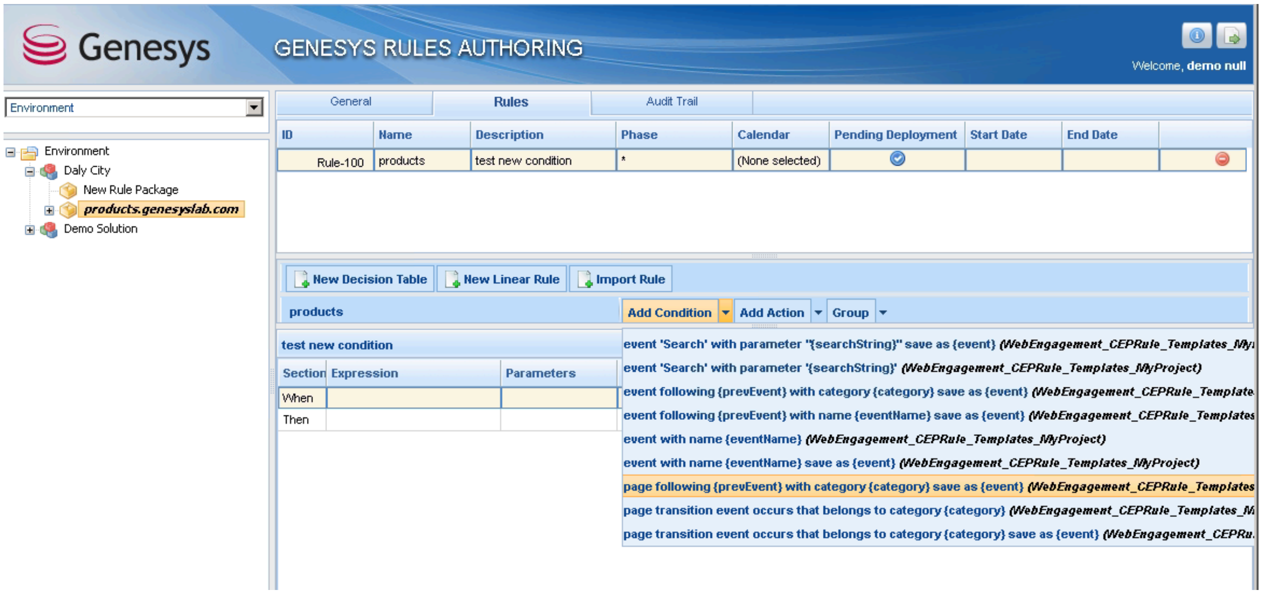
Language Expression:

```
page following {prevEvent} with category {category} save as {event}
```

Rule Language Mapping:

```
{event}: Event(eval({event}.getName().equals('PageEntered')) && eval({event}.hasCategory('{category}')), this after {prevEvent})
```

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



ID	Name	Description	Phase	Calendar	Pending Deployment	Start Date	End Date	
Rule-100	products	test new condition	*	(None selected)	✓			⊖

Section	Expression	Parameters
When		
Then		

event 'Search' with parameter "{searchString}" save as {event} *(WebEngagement_CEPRule_Templates_My*

event 'Search' with parameter '{searchString}' *(WebEngagement_CEPRule_Templates_MyProject)*

event following {prevEvent} with category {category} save as {event} *(WebEngagement_CEPRule_Template*

event following {prevEvent} with name {eventName} save as {event} *(WebEngagement_CEPRule_Templates*

event with name {eventName} *(WebEngagement_CEPRule_Templates_MyProject)*

event with name {eventName} save as {event} *(WebEngagement_CEPRule_Templates_MyProject)*

page following {prevEvent} with category {category} save as {event} *(WebEngagement_CEPRule_Templates*

page transition event occurs that belongs to category {category} *(WebEngagement_CEPRule_Templates_M*

page transition event occurs that belongs to category {category} save as {event} *(WebEngagement_CEPRu*

End

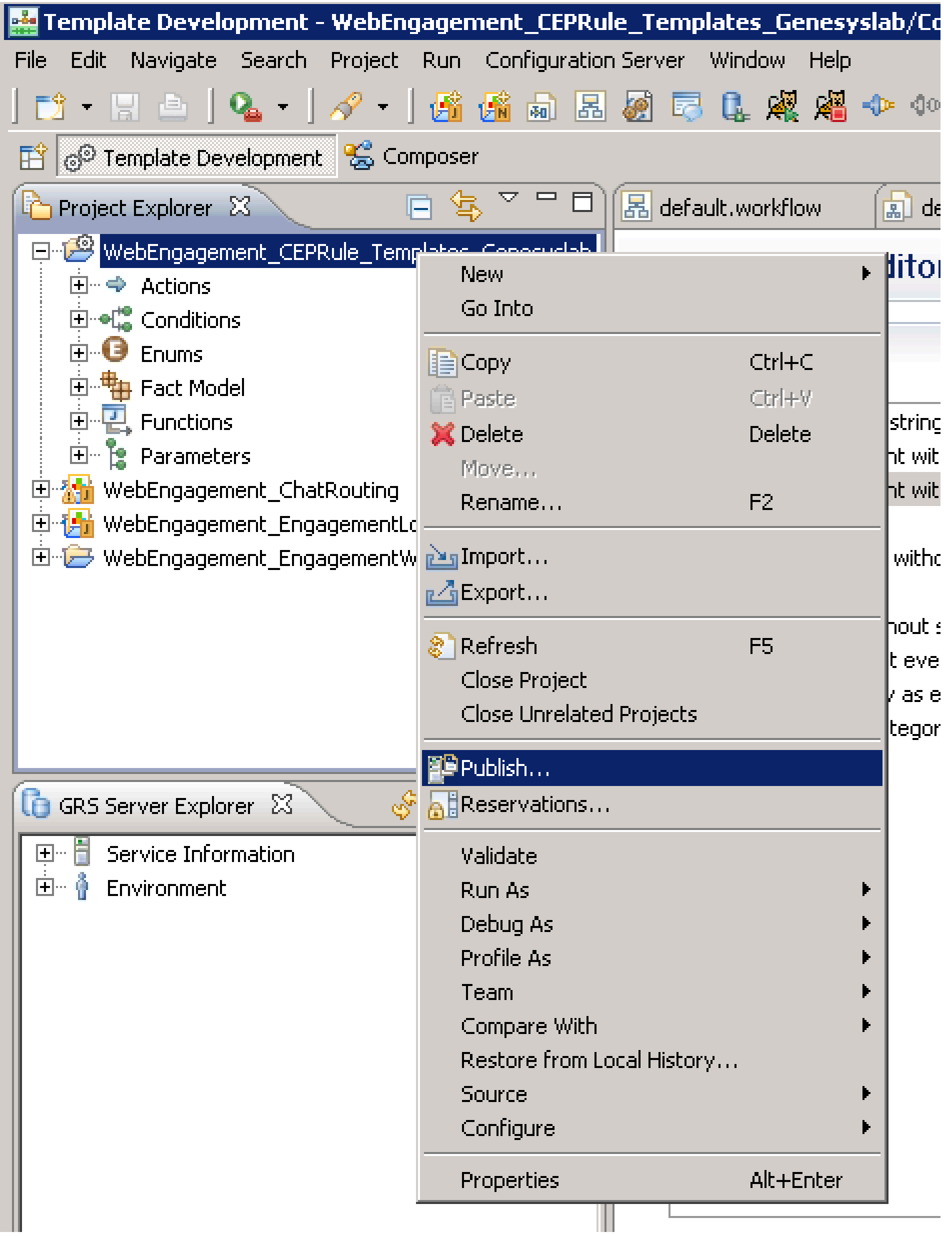
Publishing the CEP Rule Templates in the Rules Repository

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.

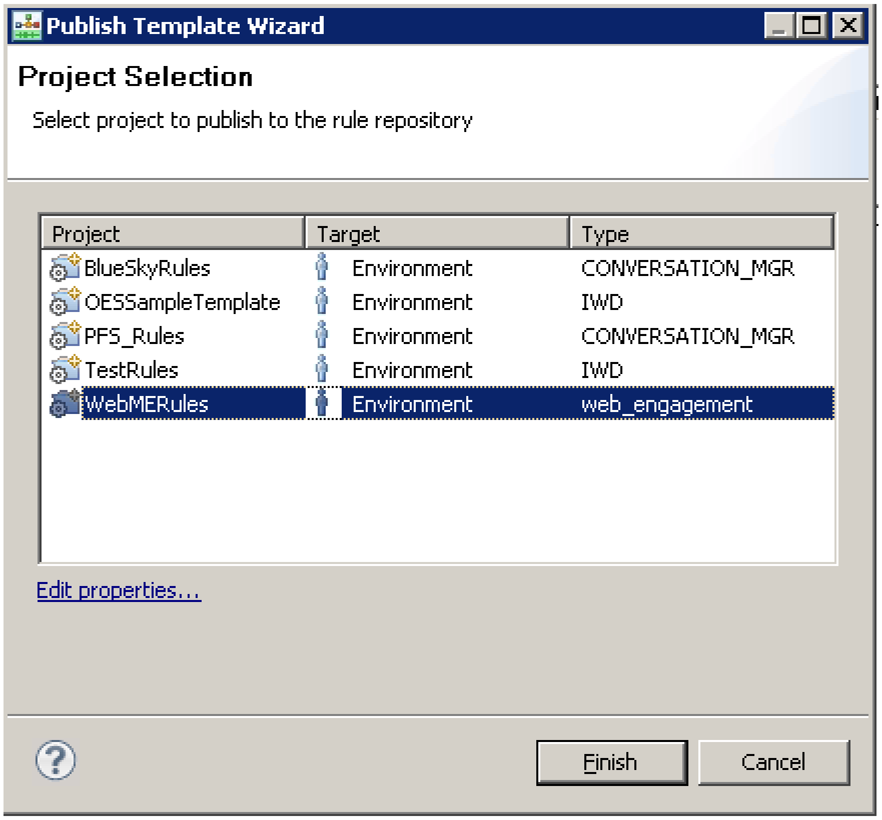
Start

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**.



Project Selection

Select project to publish to the rule repository

Project	Target	Type
BlueSkyRules	Environment	CONVERSATION_MGR
OESSampleTemplate	Environment	IWD
PFS_Rules	Environment	CONVERSATION_MGR
TestRules	Environment	IWD
WebMERules	Environment	web_engagement

Select **WebEngagement_CEP_Rule_Templates**.

4. Click **Finish**.

End

Managing Categories

Overview

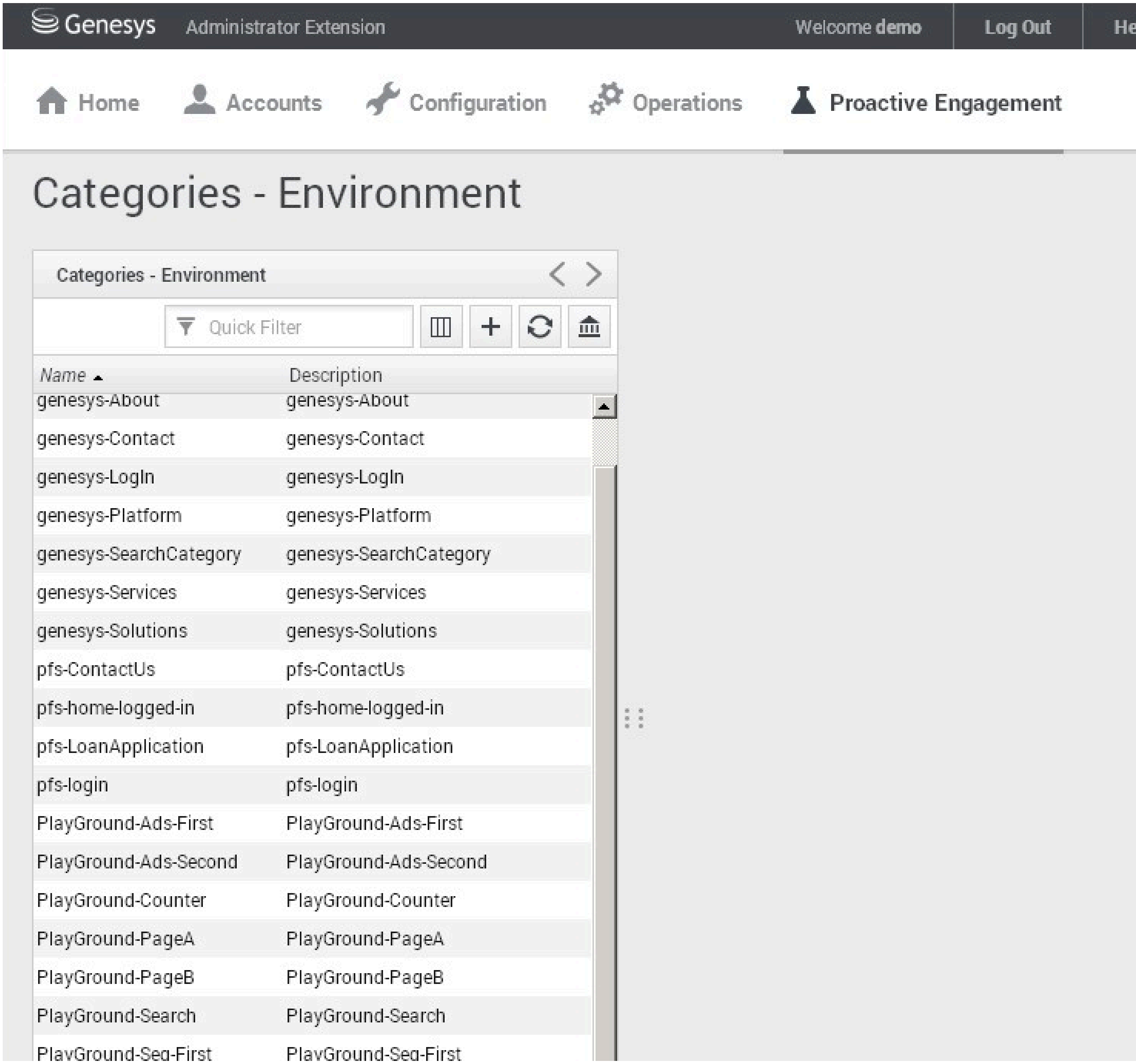
You can use the GWE Plug-in for Genesys Administrator Extension to define, in a few clicks, Web Engagement categories that contain business information related to URL or web page titles. These categories are used in the CEP rule templates, which provide rules that define when to submit actionable events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

For example, lets 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. You can add and remove categories for Web Engagement through the Category interface in the Genesys Administrator Extension plug-in. 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 `Proactive Engagement > Categories`.



Categories - Environment

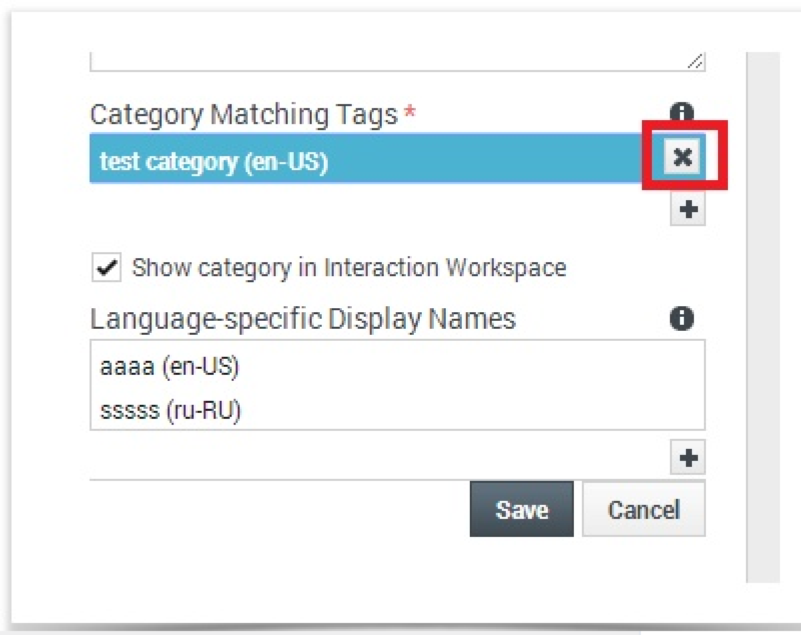
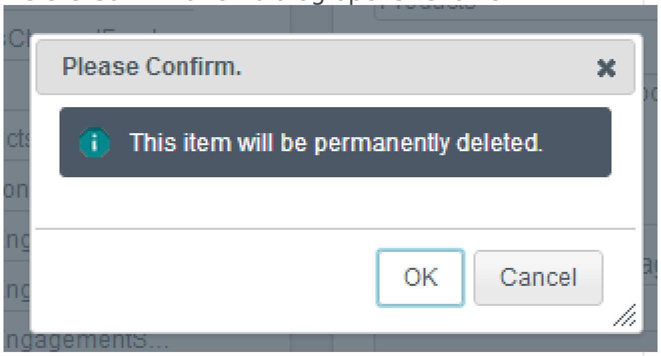
Categories - Environment

Name	Description
genesys-About	genesys-About
genesys-Contact	genesys-Contact
genesys-LogIn	genesys-LogIn
genesys-Platform	genesys-Platform
genesys-SearchCategory	genesys-SearchCategory
genesys-Services	genesys-Services
genesys-Solutions	genesys-Solutions
pfs-ContactUs	pfs-ContactUs
pfs-home-logged-in	pfs-home-logged-in
pfs-LoanApplication	pfs-LoanApplication
pfs-login	pfs-login
PlayGround-Ads-First	PlayGround-Ads-First
PlayGround-Ads-Second	PlayGround-Ads-Second
PlayGround-Counter	PlayGround-Counter
PlayGround-PageA	PlayGround-PageA
PlayGround-PageB	PlayGround-PageB
PlayGround-Search	PlayGround-Search
PlayGround-Seq-First	PlayGround-Seq-First

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.  Category Matching Tags * test category (en-US) ☑ Show category in Interaction Workspace Language-specific Display Names aaaa (en-US) sssss (ru-RU) Click `Delete`.
Delete categories.	Select the category in the list and click `Delete`. The Delete Confirmation dialog opens. Click OK.  Please Confirm. This item will be permanently deleted. 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.

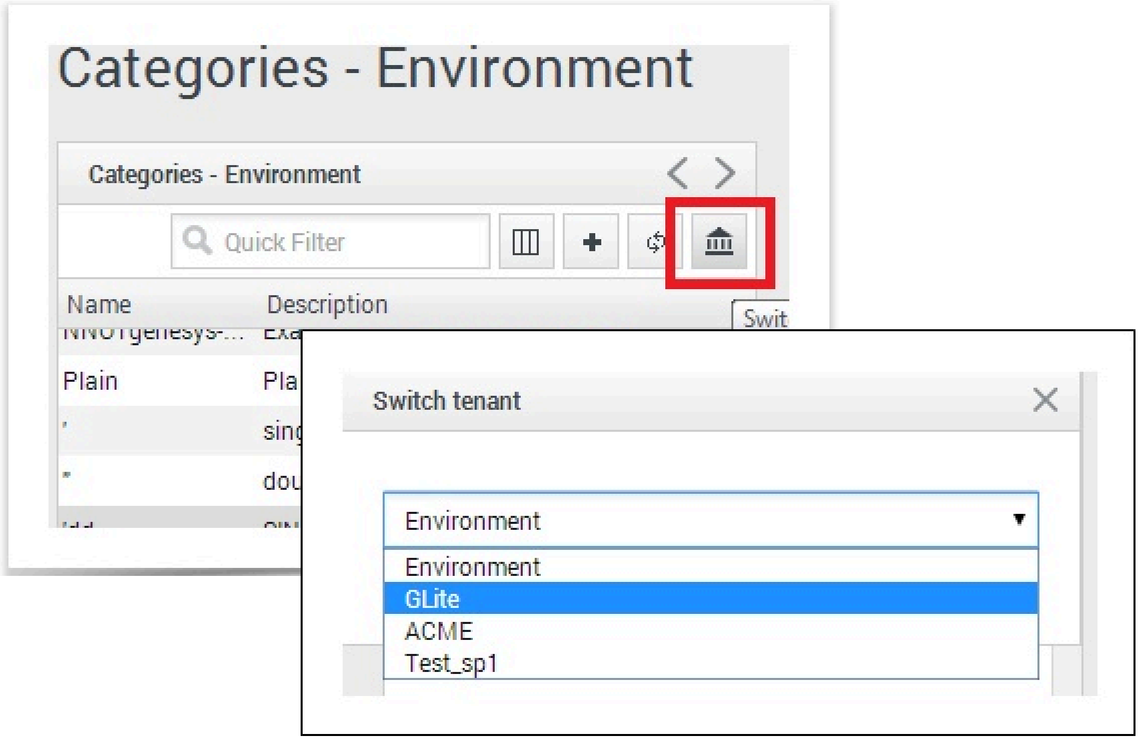
Creating a Category

Prerequisites

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

Start

1. In Genesys Administrator Extension, navigate to `Proactive 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, a tag to identify the http://www.genesyslab.com/products/genesys-inbound-voice/overview.aspx page could be the plain expression '`genesys-inbound-voice`' or the regular expression 'Inbound Voice'.

Prerequisites

- You completed Creating a Category.

Start

1. In Genesys Administrator Extension, navigate to `Proactive Engagement > Categories` and select a catetory. 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.

Field	Description
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*\_composer-project\WebEngagement_EngagementWidgets\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="TimeoutEvent10" name="Timeout-10" condition=""
postcondition="document.hasFocus() === true">
            <trigger name="TimeoutTrigger" element="" action="timer:10000" type="timeout"
url="" count="1" />
        </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 Frontend 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*\frontend\src\main\webapp\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 includes a few events to help you get started with your DSL customizations: SearchEvent, TimeoutEvent10, and TimeoutEvent30. The following sections show you how you can customize these events to work 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 id="SearchEvent" 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 id="SearchEvent" 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 id="SearchEvent" 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 two timeout events: timeout-10 and timeout-30.

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

You can customize these events or disable one or both to suit your business needs. By default, these events are triggered with a 10-second and 30-second delay after the tracking script is initialized on the page. The only difference between the events is the **action** attribute, which defines the timeout in milliseconds.

Both events have the **postcondition** attribute set to "`document.hasFocus() === true`", which checks whether the focus is on the current page. The timeout event is only triggered if the **postcondition** returns true.

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.

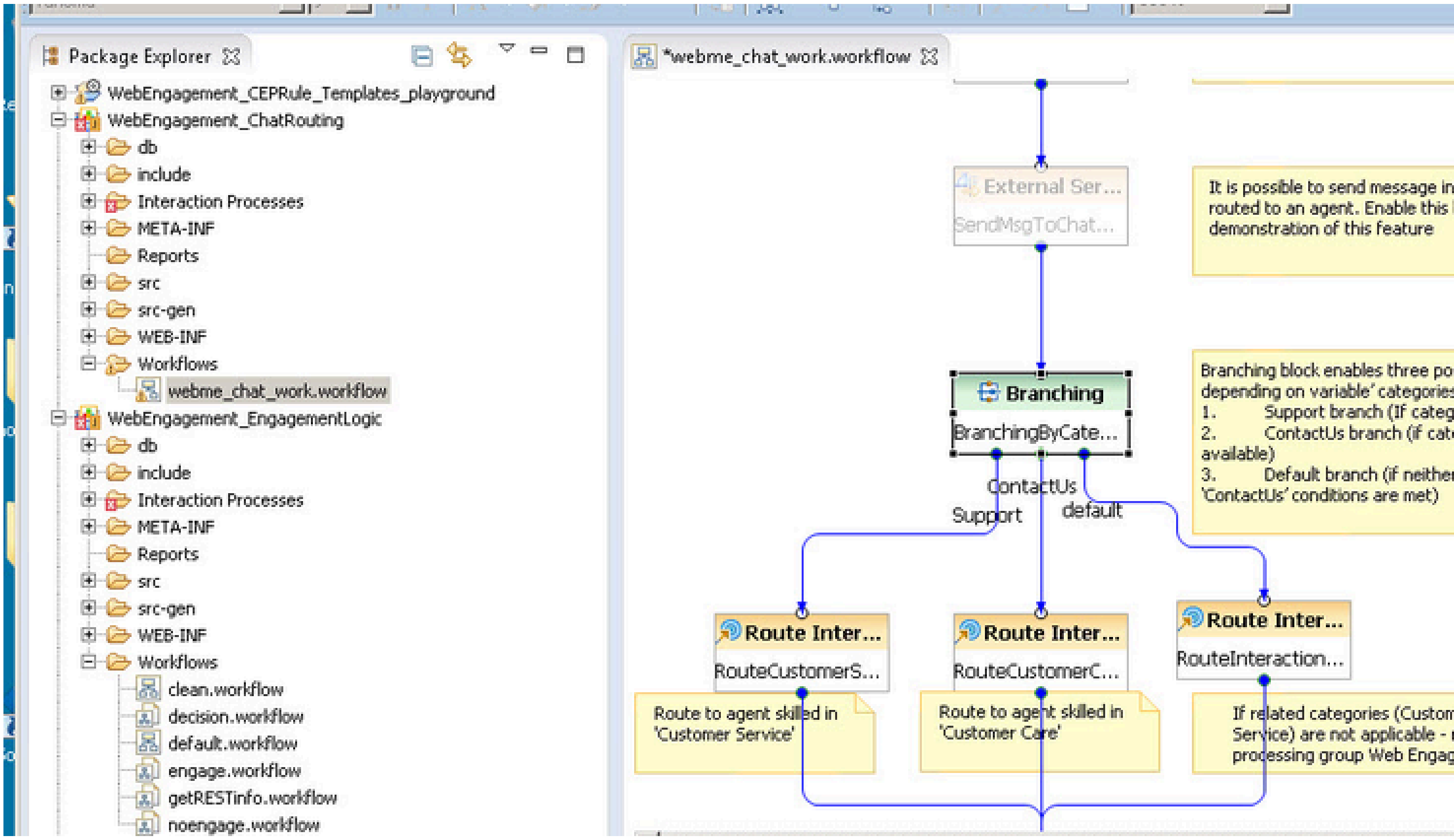
In this case, you can use the `_gt.push()` method which allows you to decide when events should be submitted and which data they generate, directly from your web pages. See Monitoring JS API Reference for further details.

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*\_composer-project\** 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:



It is possible to send message in
routed to an agent. Enable this
demonstration of this feature

Branching block enables three po
depending on variable' categories
1. Support branch (If categ
2. ContactUs branch (if cate
available)
3. Default branch (if neither
'ContactUs' conditions are met)

Route to agent skilled in
'Customer Service'

Route to agent skilled in
'Customer Care'

If related categories (Custom
Service) are not applicable -
processing group Web Engag

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.genesy.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 Frontend server by setting the cep.domainSeparation option to `true`.

Creating a Rules Package

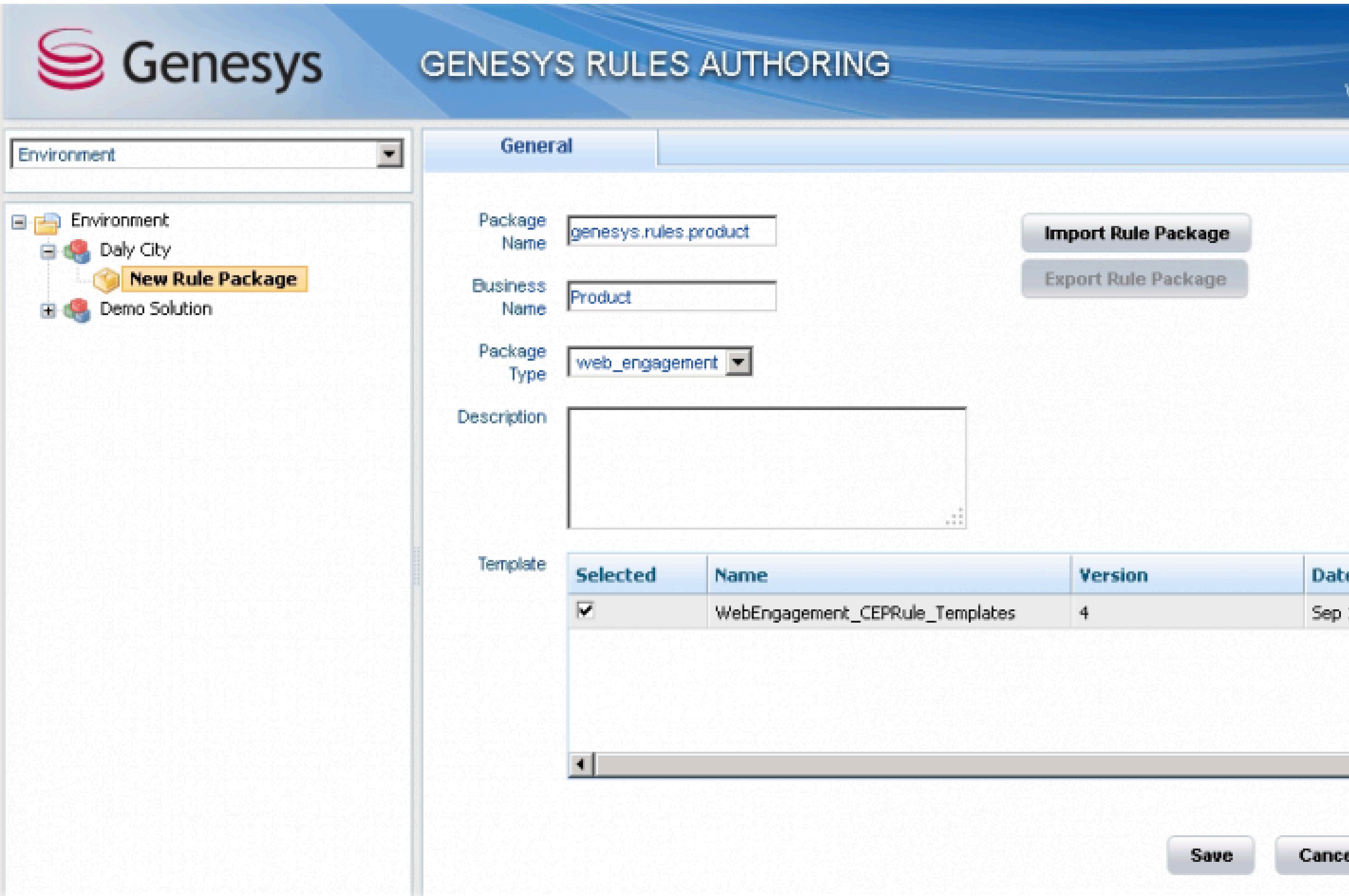
Complete the steps below to create the rules package associated with your Web Engagement application. This procedure is an example of how to create a rules package. For further information about creating rules, refer to the Genesys Rules System Deployment Guide.

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.



Package Name: genesys.rules.product

Business Name: Product

Package Type: web_engagement

Description:

Template:

Selected	Name	Version	Dat
☑	WebEngagement_CEPRule_Templates	4	Sep

GWENewPackageGRA.PNG

4. Click Save.

End

Creating Rules in the Rules Package

Prerequisites

- Creating a Rules Package

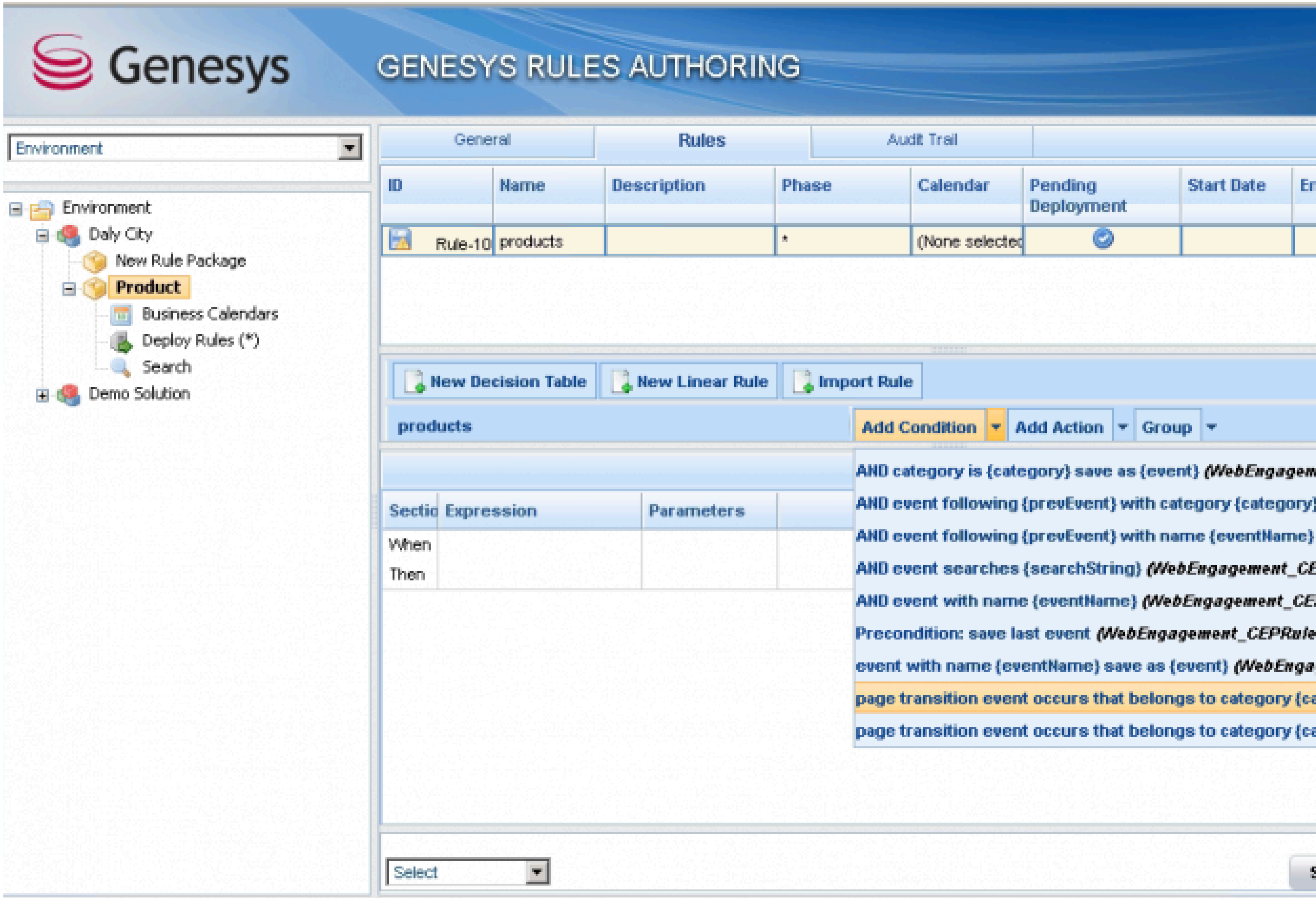
Start

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.

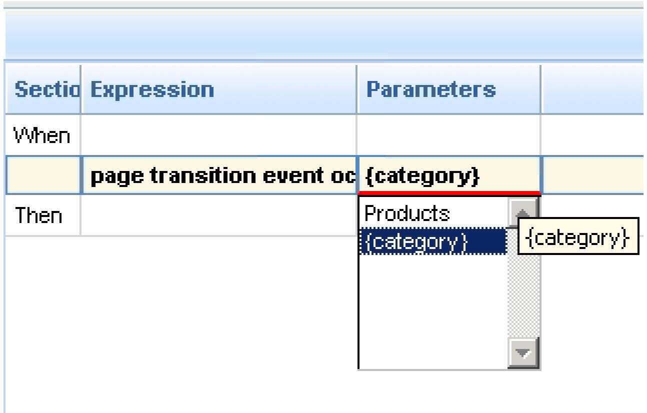

Genesys
GENESYS RULES AUTHORING
Environment
Environment
Daly City
New Rule Package
Product
Business Calendars
Deploy Rules (*)
Search
Demo Solution
General
Rules
Audit Trail
New Decision Table
New Linear Rule
Import Rule
products
Add Condition
Add Action
Group
AND category is {category} save as {event} (WebEngagem
AND event following {prevEvent} with category {category}
AND event following {prevEvent} with name {eventName}
AND event searches {searchString} (WebEngagement_CE
AND event with name {eventName} (WebEngagement_CE
Precondition: save last event (WebEngagement_CEPRule
event with name {eventName} save as {event} (WebEnga
page transition event occurs that belongs to category {ca
page transition event occurs that belongs to category {ca
Select

ID	Name	Description	Phase	Calendar	Pending Deployment	Start Date	En
Rule-10	products		*	(None selected	✓		

Sectio	Expression	Parameters
When		
Then		

Select your rule's condition.

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


Sectio	Expression	Parameters
When		
	page transition event oc	{category}
Then		

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

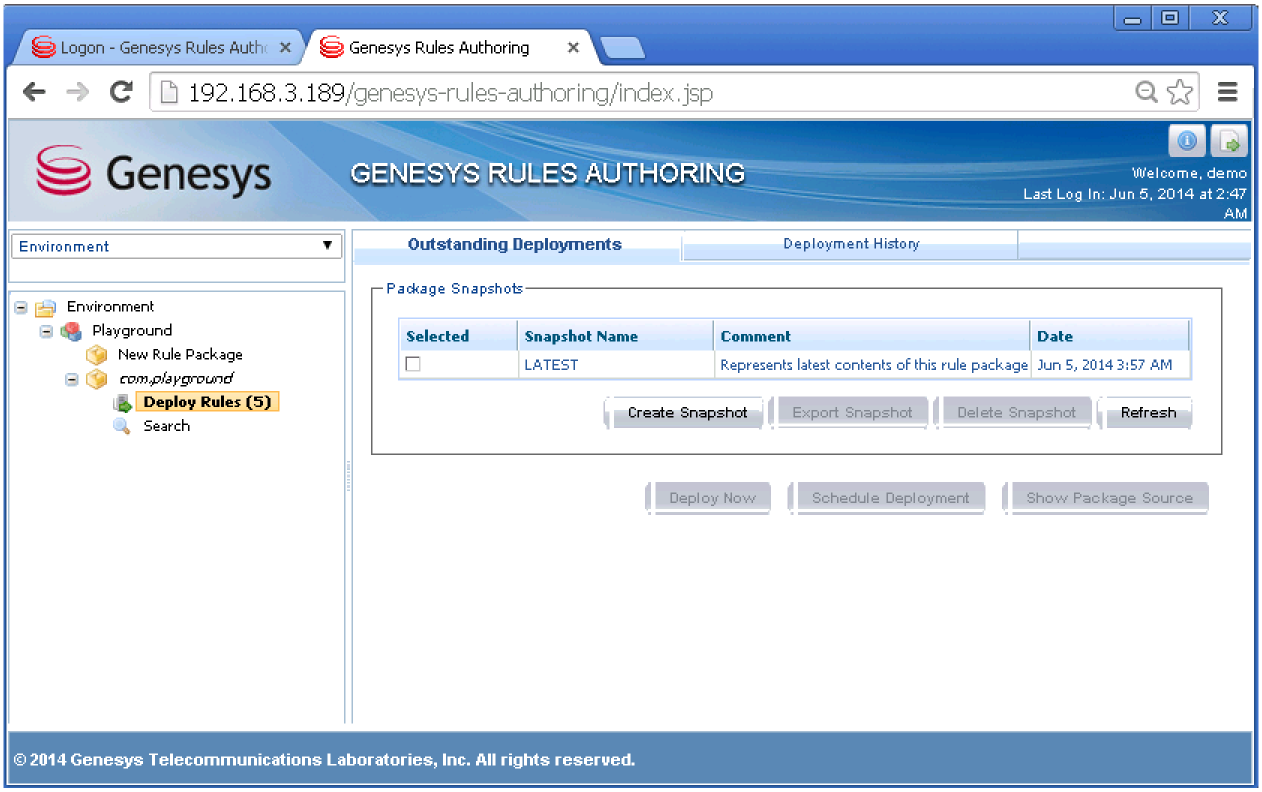
Deploying the Rules Package

Prerequisites

- You started the Web Engagement servers.

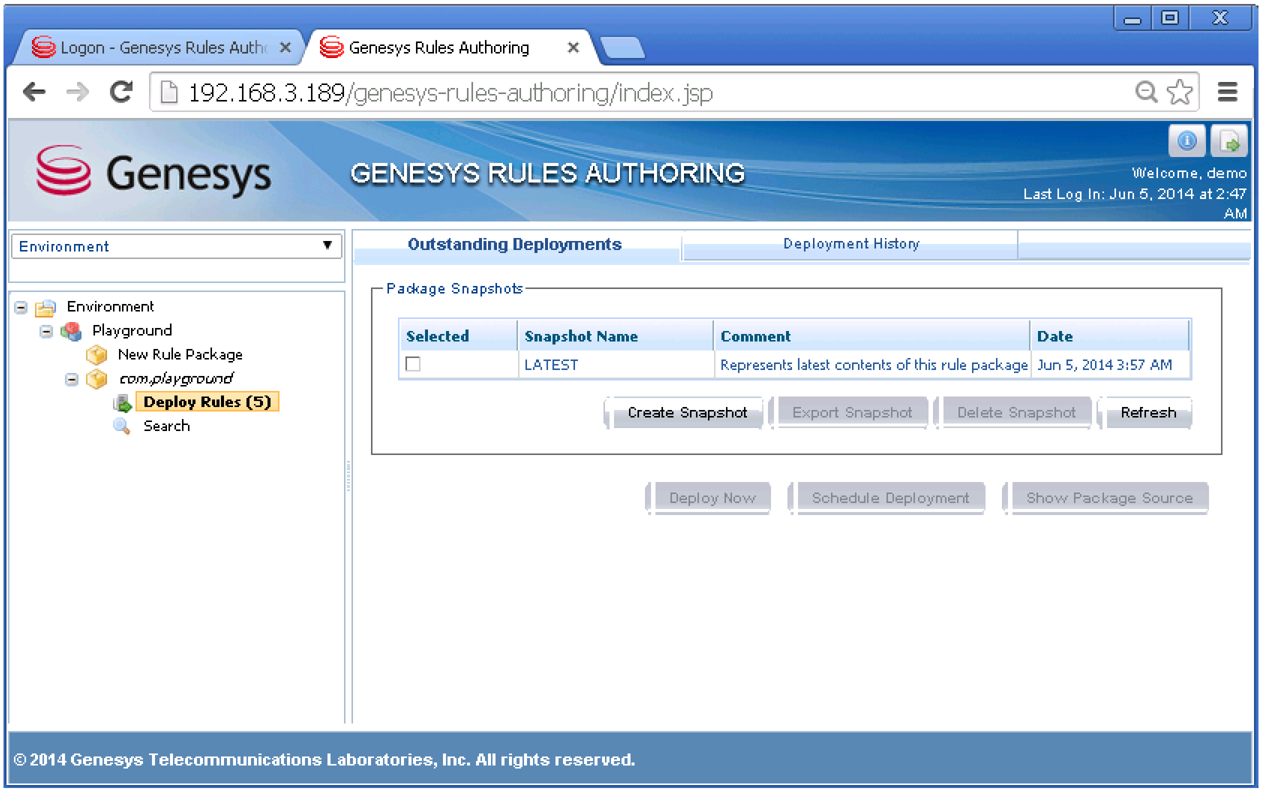
Start

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



Selected	Snapshot Name	Comment	Date
□	LATEST	Represents latest contents of this rule package	Jun 5, 2014 3:57 AM

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**.



Selected	Snapshot Name	Comment	Date
□	LATEST	Represents latest contents of this rule package	Jun 5, 2014 3:57 AM

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.

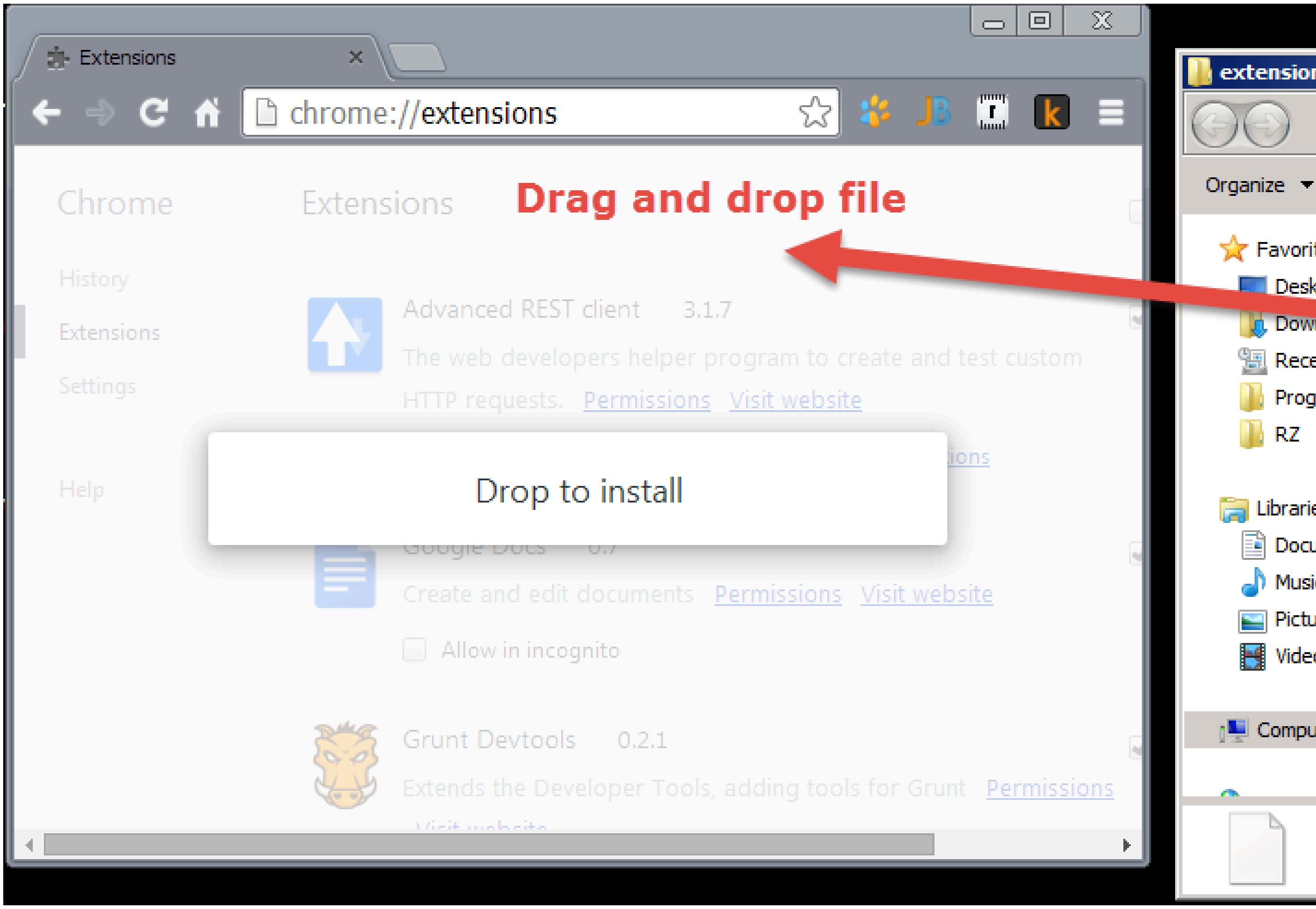
Chromium (Windows, Mac, Linux) and Chrome (Mac, Linux)

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

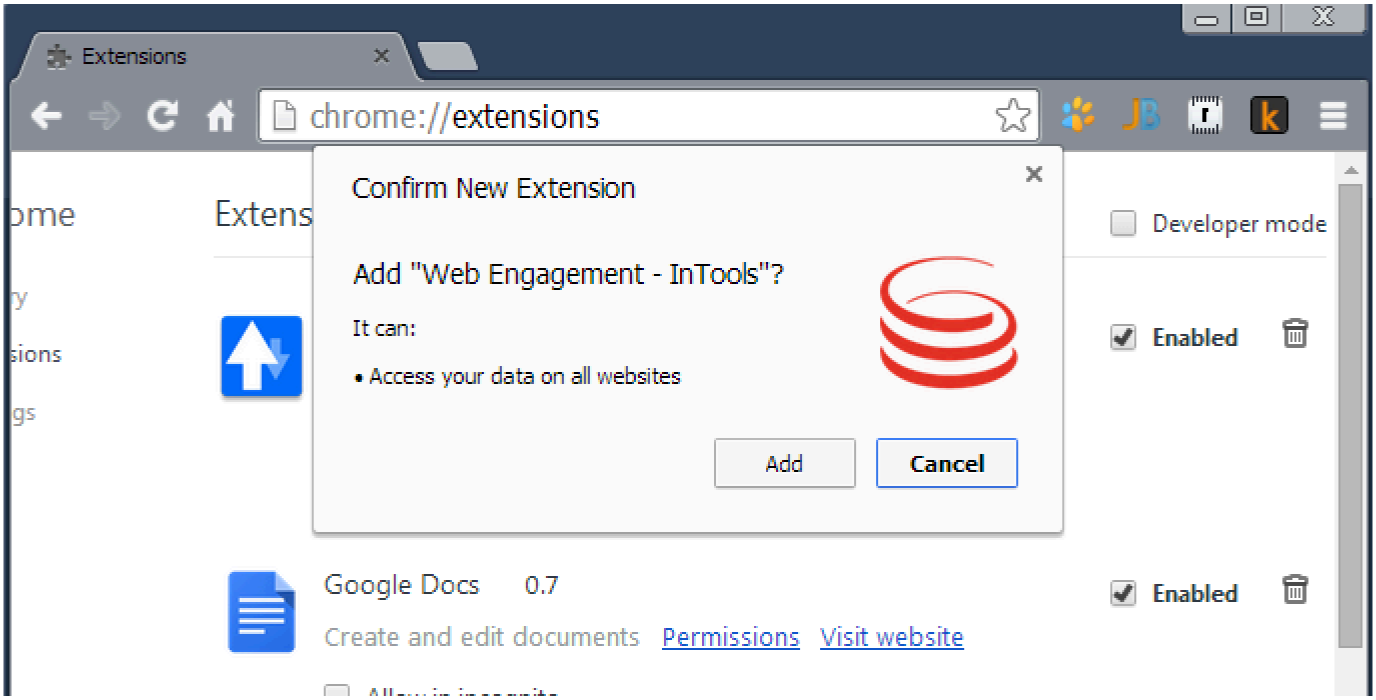
> **Important**
>
> Chrome in Windows has some restrictions about installing extensions that are not from the Chrome Store. If Chrome blocks you from installing InTools in Windows, you must install the unpacked extension in developer mode. See Chrome (Windows) for installation instructions.

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.


Extensions
chrome://extensions
Chrome
Extensions
Drag and drop file
History
Extensions
Settings
Help
Advanced REST client 3.1.7
The web developers helper program to create and test custom HTTP requests. Permissions Visit website
Drop to install
Google Docs 0.7
Create and edit documents Permissions Visit website
Allow in incognito
Grunt Devtools 0.2.1
Extends the Developer Tools, adding tools for Grunt Permissions
extension
Organize
Favorit
Desk
Dow
Rece
Prog
RZ
Librarie
Docu
Musi
Pictu
Videc
Compu

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



Confirm New Extension

Add "Web Engagement - InTools"?

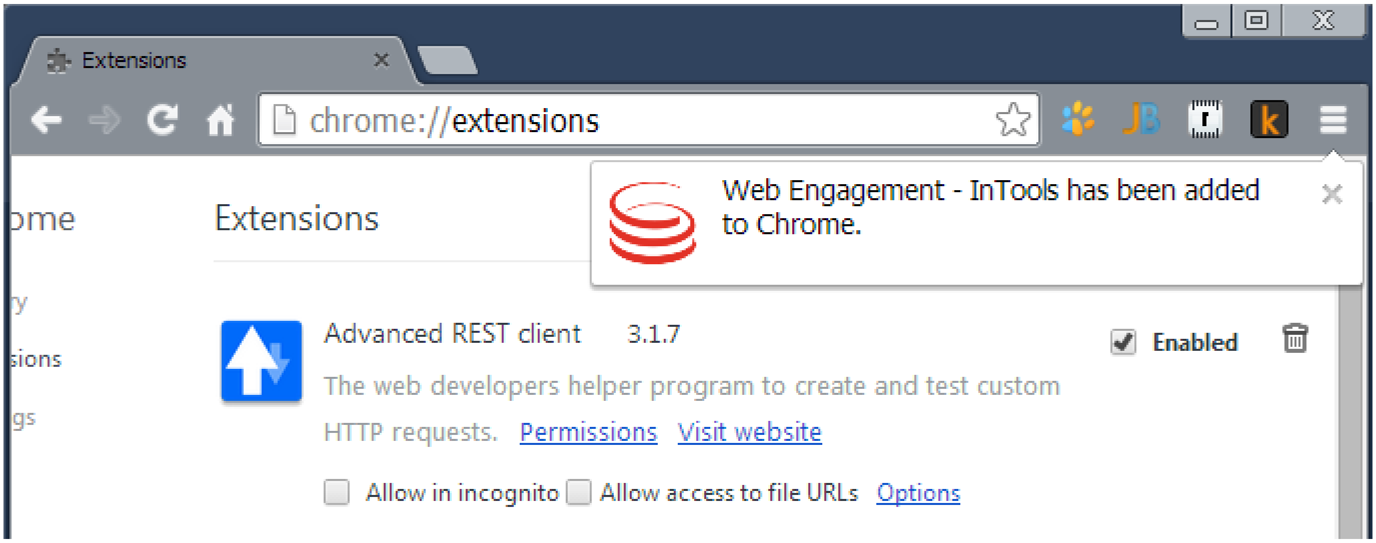
It can:

- Access your data on all websites

Google Docs 0.7

Create and edit documents Permissions Visit website

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



Web Engagement - InTools has been added to Chrome.

Advanced REST client 3.1.7

The web developers helper program to create and test custom HTTP requests. Permissions Visit website

□ Allow in incognito □ Allow access to file URLs

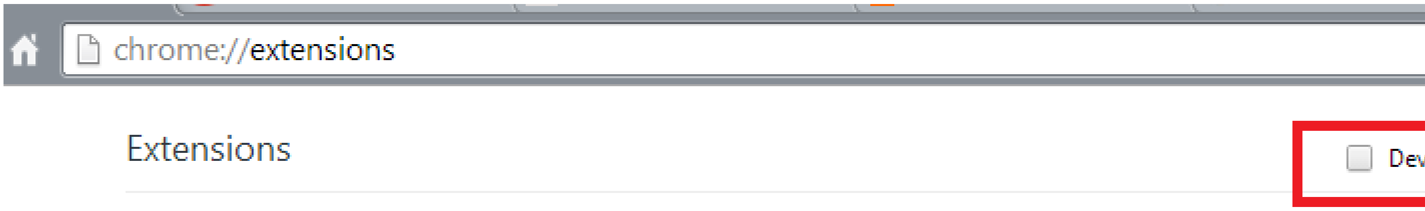
.

End

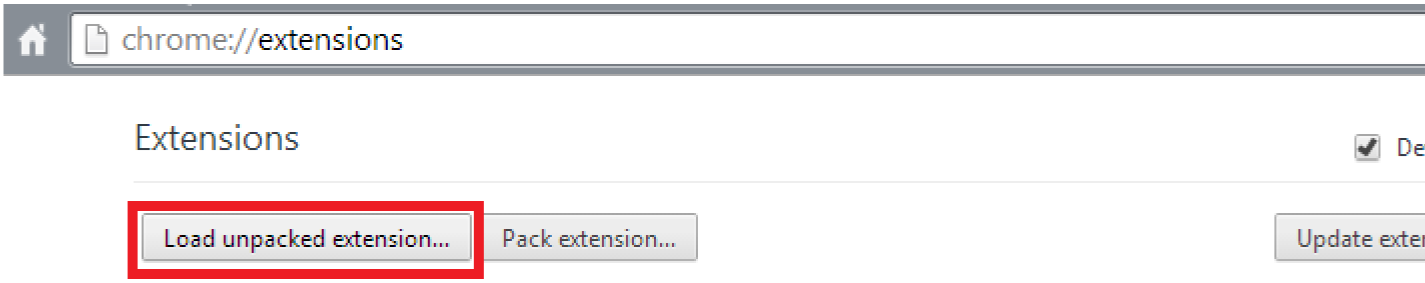
Chrome (Windows)

1. Run the Chrome browser.

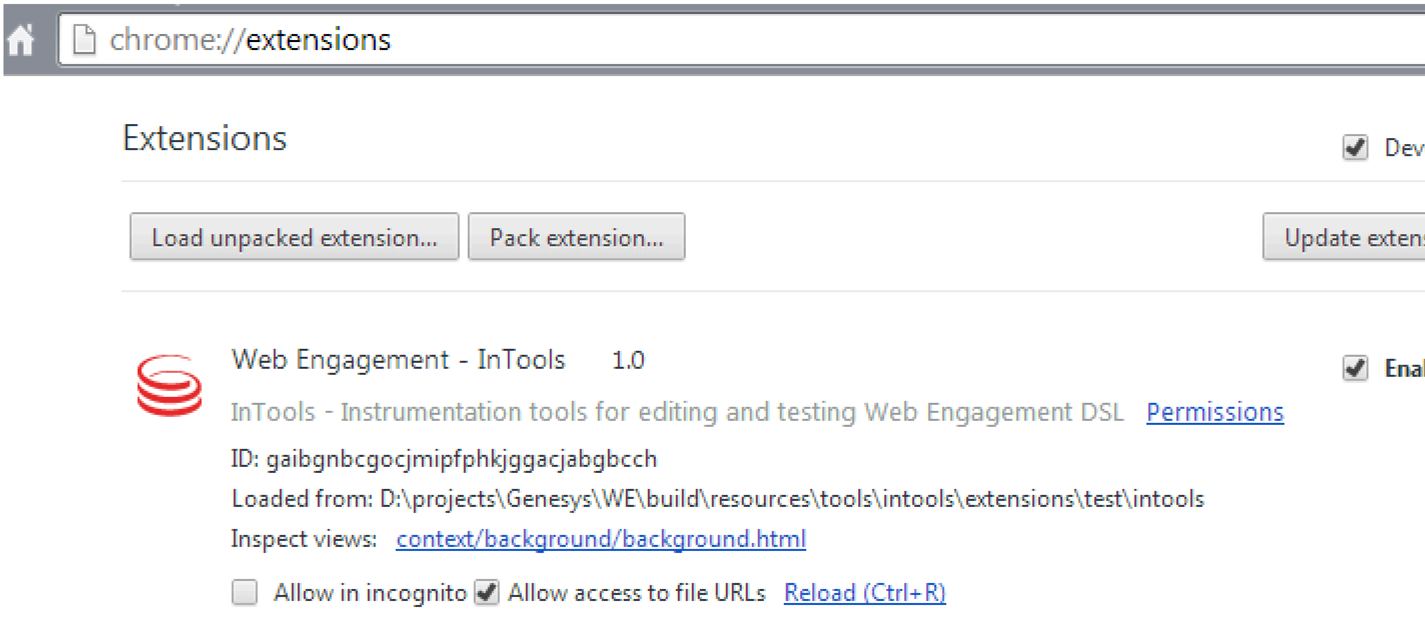
2. Navigate to the **chrome://extensions** page.
3. Enable Developer Mode.



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:



Web Engagement - InTools 1.0

InTools - Instrumentation tools for editing and testing Web Engagement DSL Permissions

ID: gaibgnbcgocjmipfphkjggacjabgbcch

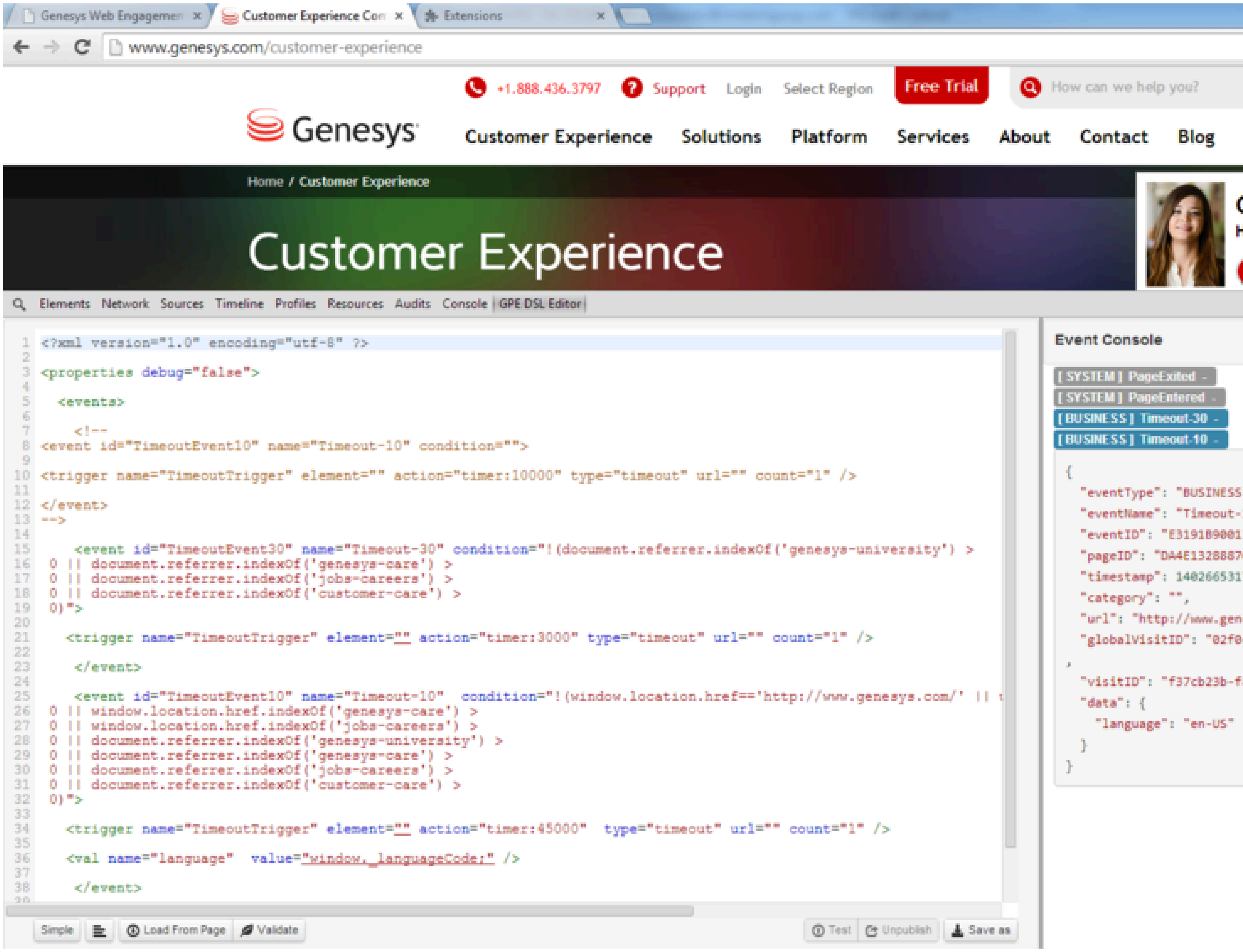
Loaded from: D:\projects\Genesys\WE\build\resources\tools\intools\extensions\test\intools

Inspect views: context/background/background.html

□ Allow in incognito ☑ Allow access to file URLs Reload (Ctrl+R)

GWE DSL Editor

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



```
<?xml version="1.0" encoding="utf-8" ?>

<properties debug="false">

  <events>

    <!--
<event id="TimeoutEvent10" name="Timeout-10" condition="">

<trigger name="TimeoutTrigger" element="" action="timer:10000" type="timeout" url="" count="1" />

</event>
-->

    <event id="TimeoutEvent30" name="Timeout-30" condition="!(document.referrer.indexOf('genesys-university') >
 0 || document.referrer.indexOf('genesys-care') >
 0 || document.referrer.indexOf('jobs-careers') >
 0 || document.referrer.indexOf('customer-care') >
 0)">

   <trigger name="TimeoutTrigger" element="" action="timer:3000" type="timeout" url="" count="1" />

    </event>

    <event id="TimeoutEvent10" name="Timeout-10"  condition="!(window.location.href=='http://www.genesys.com/' ||
 0 || window.location.href.indexOf('genesys-care') >
 0 || window.location.href.indexOf('jobs-careers') >
 0 || document.referrer.indexOf('genesys-university') >
 0 || document.referrer.indexOf('genesys-care') >
 0 || document.referrer.indexOf('jobs-careers') >
 0 || document.referrer.indexOf('customer-care') >
 0)">

   <trigger name="TimeoutTrigger" element="" action="timer:45000"  type="timeout" url="" count="1" />

   <val name="language"  value="window._languageCode;" />

    </event>
```

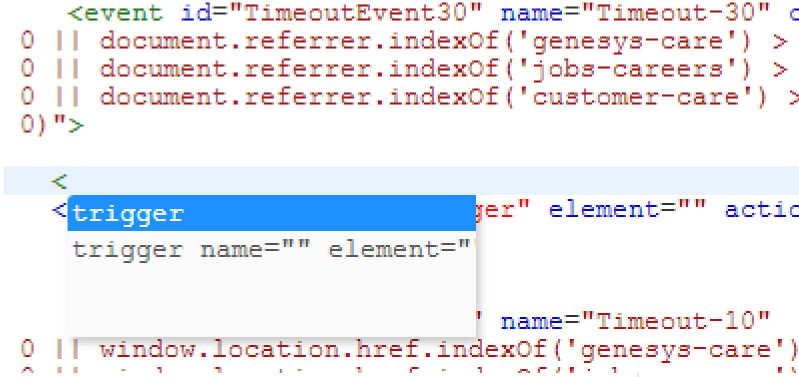
```
{
  "eventType": "BUSINESS
  "eventName": "Timeout-
  "eventID": "E3191B9001
  "pageID": "DA4E1328887
  "timestamp": 140266531
  "category": "",
  "url": "http://www.gen
  "globalVisitID": "02f0
,
  "visitID": "f37cb23b-f
  "data": {
    "language": "en-US"
  }
}
```

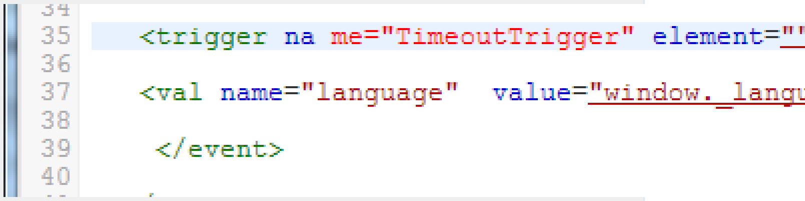
The DSL Editor

Advanced Editor

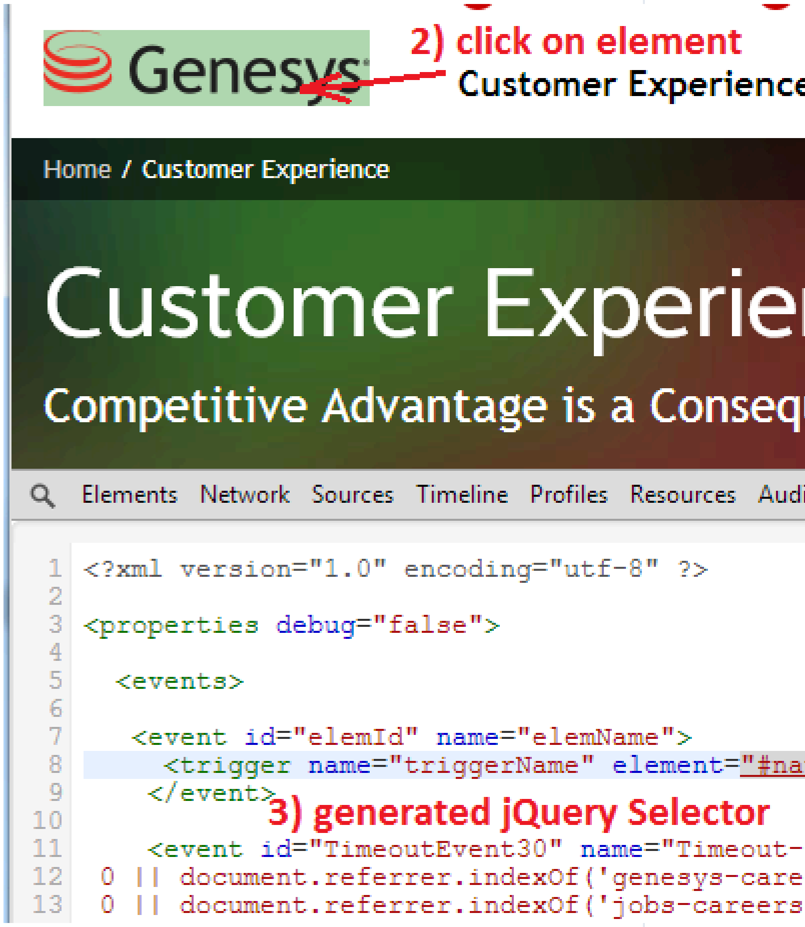
The Advanced Editor is a text editor for Genesys Web Engagement DSL that provides the following features:

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.
	
	`<event id="TimeoutEvent30" name="Timeout-30" c`
	`0 || document.referrer.indexOf('genesys-care') >`
	`0 || document.referrer.indexOf('jobs-careers') >`
	`0 || document.referrer.indexOf('customer-care') >`
	`0)">`
	`<`
	`<trigger ger" element="" actic`
	`" name="Timeout-10"`
	`0 || window.location.href.indexOf('genesys-care')`
	
	The code complete menu.

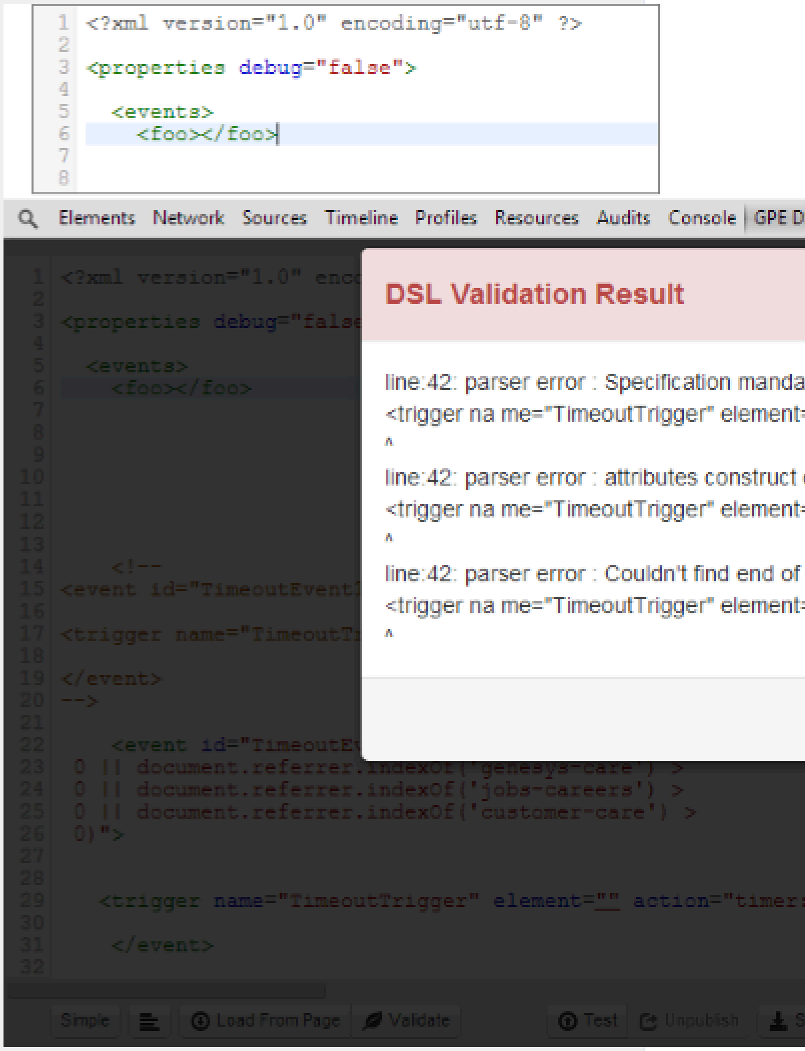
XML syntax validation.	Automatic.
	
	`<trigger na me="TimeoutTrigger" element=""`
	`<val name="language" value="window._langu`
	`</event>`
	Syntax validation

Feature	Usage

jQuery selector highlighting. The tool can generate values for underscored attributes ("element" and "value") by clicking HTML elements on the page.	 2) click on element
	Genesys
	Customer Experience
	Home / Customer Experience
	Customer Experien
	Competitive Advantage is a Conseq
	Elements Network Sources Timeline Profiles Resources Audi
	3) generated jQuery Selector
	`<?xml version="1.0" encoding="utf-8" ?>`
	`<properties debug="false">`
	`<events>`
	`<event id="elemId" name="elemName">`
	`<trigger name="triggerName" element="#na`
	`</event>`
	`<event id="TimeoutEvent30" name="Timeout-`
	`0 || document.referrer.indexOf('genesys-care`
	`0 || document.referrer.indexOf('jobs-careers`
	Selector highlighting

DSL structure validation. The tool validates the names and positions of elements, including children.	Click Validate.

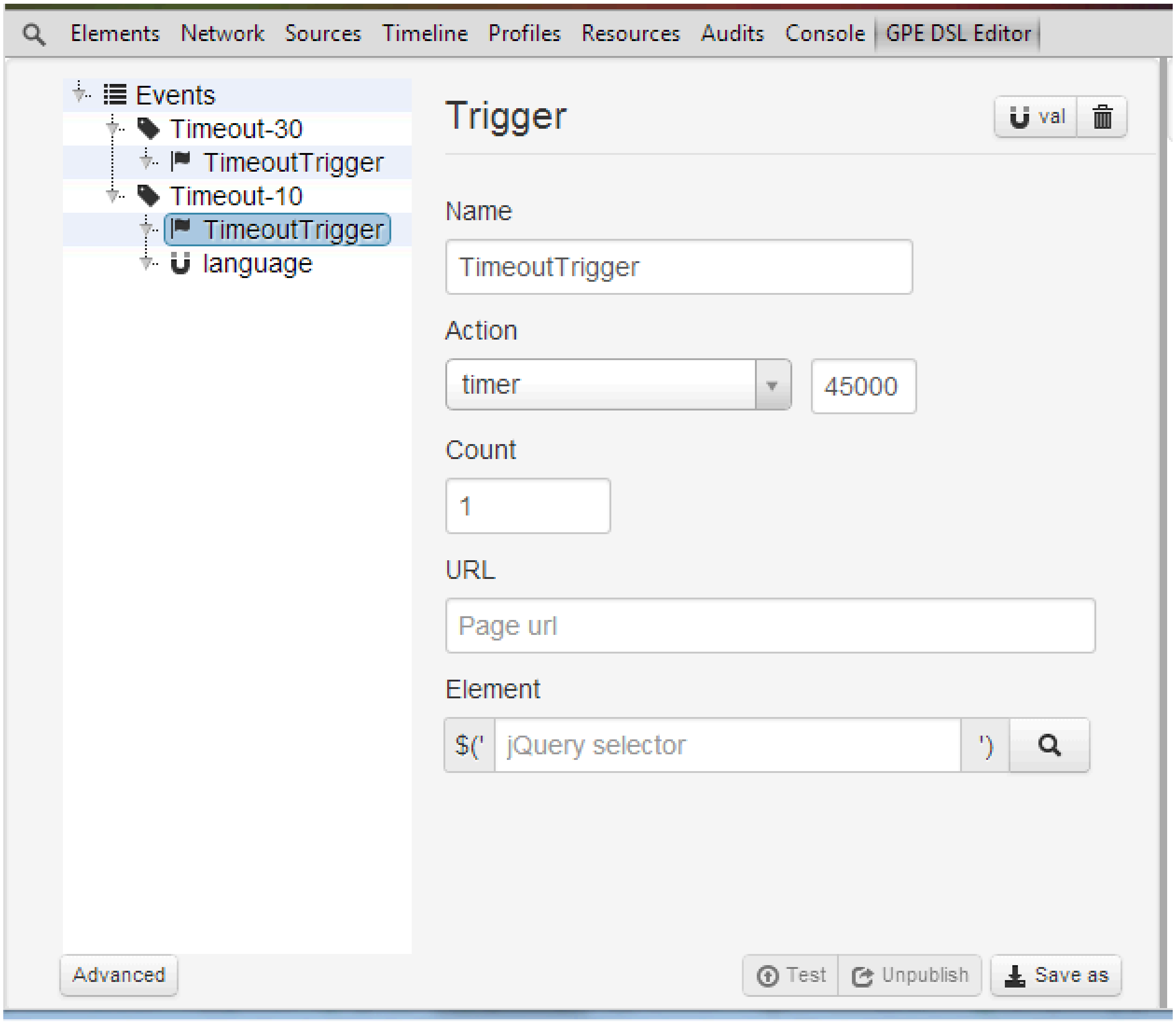
Feature	Usage

	
	`<?xml version="1.0" encoding="utf-8" ?>`
	
	`<properties debug="false">`
	
	`  <events>`
	`    <foo></foo>`
	
	DSL Validation Result
	line:42: parser error : Specification manda
	`<trigger na me="TimeoutTrigger" element=`
	^
	line:42: parser error : attributes construct
	`<trigger na me="TimeoutTrigger" element=`
	^
	line:42: parser error : Couldn't find end of
	`<trigger na me="TimeoutTrigger" element=`
	^
	 Elements
	Network
	Sources
	Timeline
	Profiles
	Resources
	Audits
	Console
	GPE D
	Simple
	Load From Page
	Validate
	Test
	Unpublish
	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.



Trigger

Name

TimeoutTrigger

Action

timer

45000

Count

1

URL

Page url

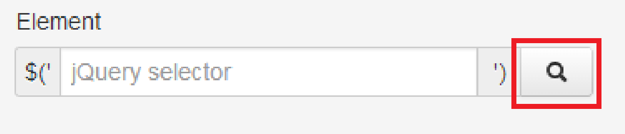
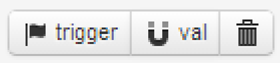
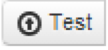
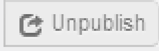
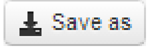
Element

$(' jQuery selector ')

The Simple Editor

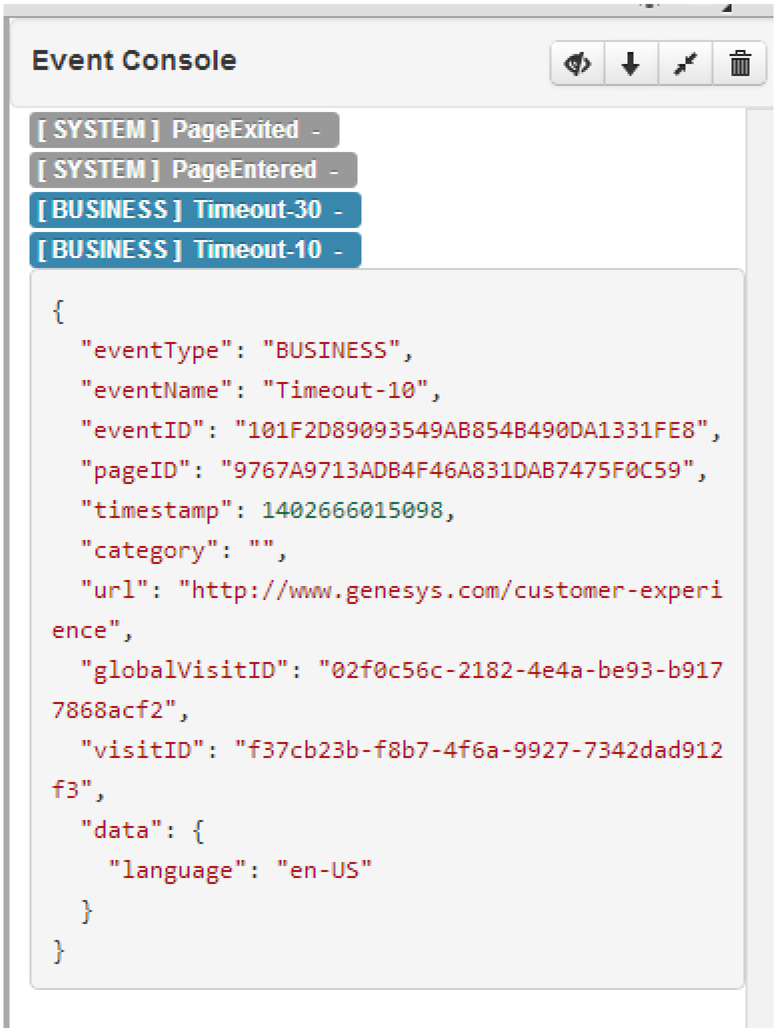
The editor includes the following features:

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

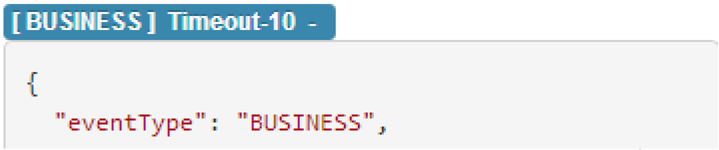
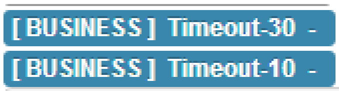
The Event Console monitors the generated events (both system and business). The event body includes all the event data in JSON format. The same format is posted to the Frontend Server.



```
{
  "eventType": "BUSINESS",
  "eventName": "Timeout-10",
  "eventID": "101F2D89093549AB854B490DA1331FE8",
  "pageID": "9767A9713ADB4F46A831DAB7475F0C59",
  "timestamp": 1402666015098,
  "category": "",
  "url": "http://www.genesys.com/customer-experi
ence",
  "globalVisitID": "02f0c56c-2182-4e4a-be93-b917
7868acf2",
  "visitID": "f37cb23b-f8b7-4f6a-9927-7342dad912
f3",
  "data": {
    "language": "en-US"
  }
}
```

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.  [BUSINESS] Timeout-10 - { "eventType": "BUSINESS", The full event information is displayed.  [BUSINESS] Timeout-30 - [BUSINESS] Timeout-10 - 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 ⤡.

Reporting

Genesys Web Engagement includes standard templates you can use for real-time interaction reporting. These templates report on the two kinds of interactions that GWE produces:

- **Pre-engagement phase interactions** are a specialized type of Open Media interaction called `webengagement`.
- **Engagement phase interactions** are chat or web callback interactions, depending on the selected channel.

To learn more about how to import the standard reporting templates into CCPulse+, see Standard Reporting Templates. Note that you can also create customized reports from the statistical data produced by Genesys Web Engagement.

Pre-engagement Phase Reports

Web Engagement includes the following reports based on pre-engagement phase interactions:

- Qualified `webengagement` interactions — The number of rules that are triggered on your website.
- Engaged `webengagement` interactions — The number of qualified interactions that meet the engagement strategy requirements.
- Accepted `webengagement` interactions — The number of engaged interactions that were accepted when a site visitor clicked the "accept" button on the engagement invite.

> **Warning**
>
> Accepted interactions do not always start a chat or voice engagement interaction. For example, a site visitor may be redirected to a registration form after accepting an engagement invite. If the visitor closes this form, the engagement interaction will not start.

- Canceled `webengagement` interactions — The number interactions that are rejected when a site visitor clicked the "reject" button on the engagement invite.

> **Warning**
>
> If an engagement invite is closed because it reaches the timeout limit, Web Engagement does not consider this a cancelled `webengagement` interaction. This is done to exclude cases where the site visitor never sees the engagement invitation — for example, when the visitor has multiple tabs open and the engagement invitation is not on the current tab.

- Failed `webengagement` interactions — The number of failed interactions. This includes both engaged and qualified interactions that failed because of any number of reasons, such as:

- The interaction does not meet the engagement strategy requirements. For example, there may be no agents available to handle the interaction.
- The interaction was triggered by a rule in test mode only.
- The interaction timed out.

> **Important**
>
> You cannot filter Open Media interactions in Stat Server, which restricts some types of reports. For example, you cannot report the percentage of successful engagement attempts out of the total count of qualified Open Media `webengagement` interactions.

Engagement Phase Reports

You can create engagement phase reports based on the number of processed interactions and the information contained within those interactions. The following are some engagement phase reports that can be collected from Web Engagement:

- The number of chat or web callback interactions that are triggered by Web Engagement and processed by an agent or group of agents.
- The percentage of chat or web callback interactions that are triggered by Web Engagement and processed by an agent.
- The number of chat or web callback interactions that are triggered by a specific Web Engagement rule and processed by an agent or group of agents.
- The percentage of chat or web callback interactions that are triggered by a specific Web Engagement rule and processed by an agent.

Standard Reporting Templates

Genesys Web Engagement provides the following standard reporting templates that can be imported into CCPulse+:

Pre-engagement Phase Templates

Web Engagement includes standard reporting templates for each Web Engagement statistics queue.

Reporting Template	Interaction Queue
WebEng_Accepted.xtpl	Webengagement_Accepted
WebEng_Engaged.xtpl	Webengagement_Engaged
WebEng_Failed.xtpl	Webengagement_Failed
WebEng_Qualified.xtpl	Webengagement_Qualified
WebEng_Rejected.xtpl	Webengagement_Rejected
WebEng_Timeout.xtpl	Webengagement_Timeout

Engagement Phase Templates

Web Engagement includes the following types of standard reporting templates for the engagement phase:

- Report templates on chat and web callback interactions triggered by the Web Engagement process and processed by an agent.
- Report templates for rule based reporting based on the data attached to chat and web callback interactions triggered by the Web Engagement process and processed by an agent.

Importing Genesys Web Engagement Sample Reporting Templates in CCPulse+

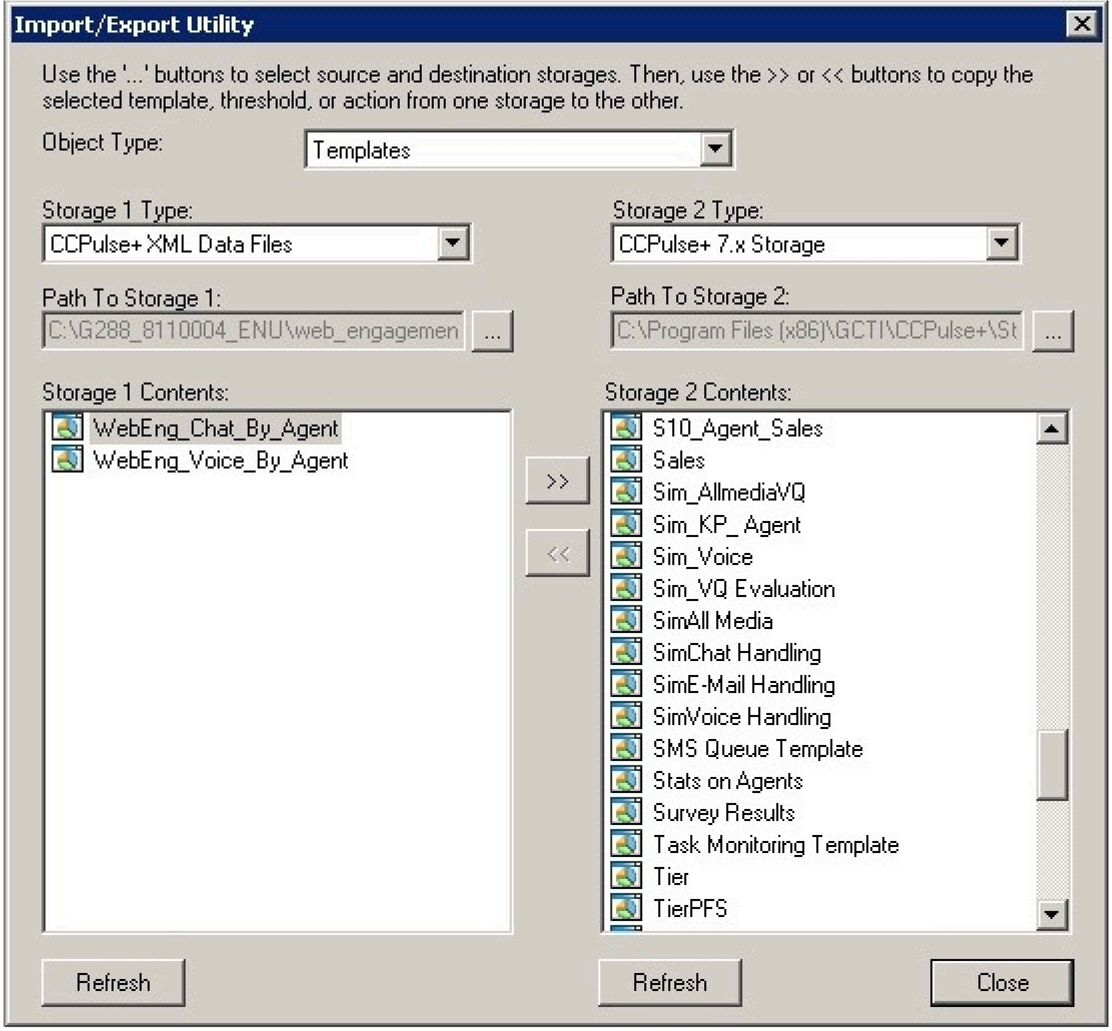
Note: For information about using CCPulse+, click `Help` in the CCPulse+ user interface or see the Reporting 8.0 CCPulse+ Administrator's Guide. The Web Engagement templates for CCPulse+ have an extension of `.xtpl`, which is an XML format. The documentation for CCPulse+ does not distinguish files by extension and refers to `.xtpl` files as XML files.

Prerequisites

- You have installed CCPulse+ version 8.0.000.36 (or later).
- You are a CCPulse+ administrator.

Start

1. Open CCPulse+ and navigate to `Tools > Import/Export`. The Import/Export Utility opens.
2. Select `Templates` for the Object Type.
3. Choose the import source.
   - For `Storage 1 Type`, select `CCPulse+ XML Data Files`.
   - For `Path To Storage 1`, click `...` and select the path to the reporting XML template(s) you wish to import. The Genesys Web Engagement Sample Reporting Templates are located in the `web_engagement_reporting_template` directory on your installation CD.
   - Click `Open`. The reporting templates are displayed in the `Storage 1 Contents` box below. For example, if you select the `Web Engagement Media Based` folder for `Path To Storage 1`, the reporting templates `WebEng_Chat_By_Agent.xtpl` and `WebEng_Voice_By_Agent.xtpl` are displayed `Storage 1 Contents`.



Import/Export Utility

Use the '...' buttons to select source and destination storages. Then, use the >> or << buttons to copy the selected template, threshold, or action from one storage to the other.

Object Type: Templates

Storage 1 Type: CCPulse+ XML Data Files

Storage 2 Type: CCPulse+ 7.x Storage

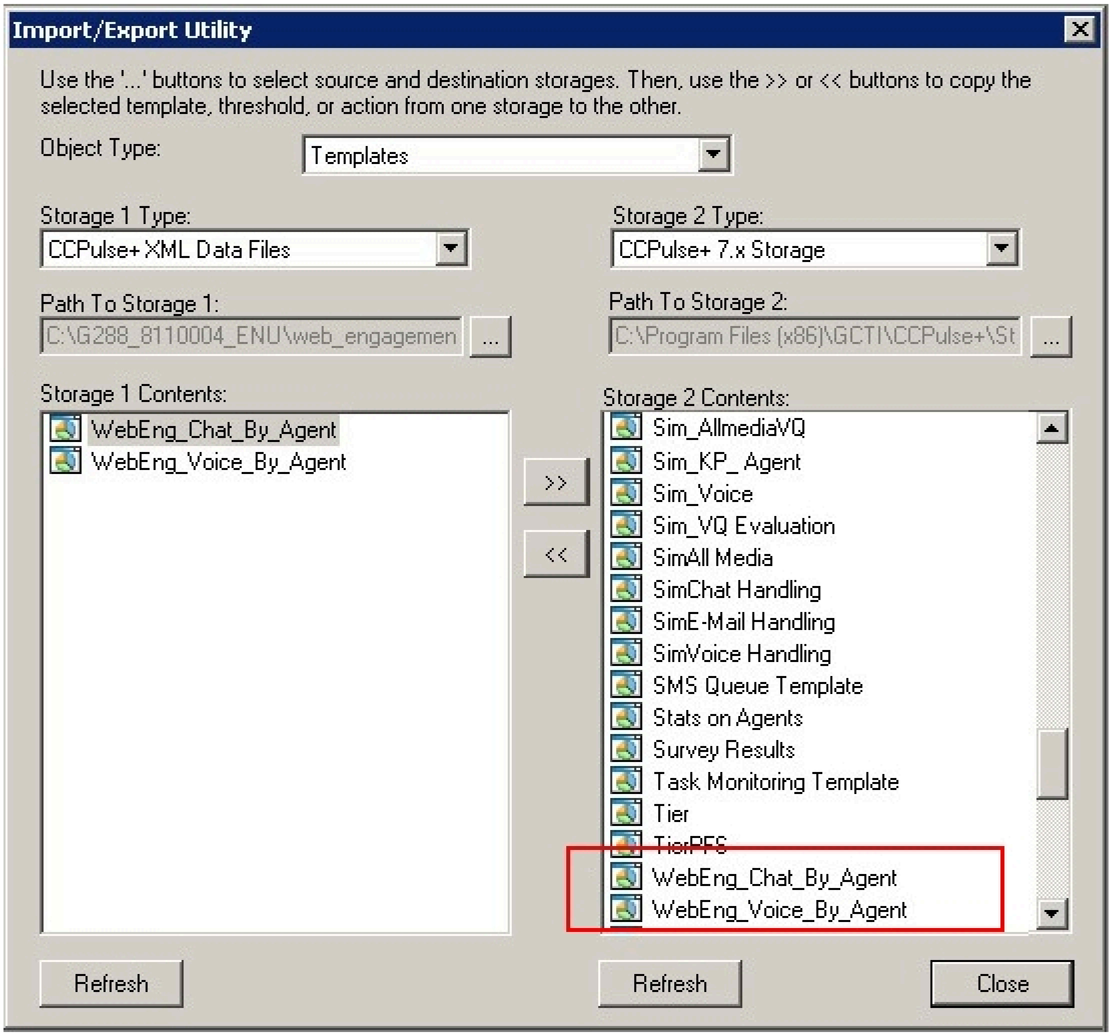
Path To Storage 1: C:\G288_8110004_ENU\web_engagemen

Path To Storage 2: C:\Program Files (x86)\GCTI\CCPulse+\St

Storage 1 Contents:
- WebEng_Chat_By_Agent
- WebEng_Voice_By_Agent

Storage 2 Contents:
- S10_Agent_Sales
- Sales
- Sim_AllmediaVQ
- Sim_KP_ Agent
- Sim_Voice
- Sim_VQ Evaluation
- SimAll Media
- SimChat Handling
- SimE-Mail Handling
- SimVoice Handling
- SMS Queue Template
- Stats on Agents
- Survey Results
- Task Monitoring Template
- Tier
- TierPFS

4. Choose the import destination.
   - Select `CCPulse+ 7.x Storage` for the `Storage 2 Type`.
   - Select a template in `Storage 1 Contents` and click the >> button. The template is copied to the `Storage 2 Contents` box. Repeat for each template you wish to copy from `Storage 1 Contents`.



Import/Export Utility

Use the '...' buttons to select source and destination storages. Then, use the >> or << buttons to copy the selected template, threshold, or action from one storage to the other.

Object Type: Templates

Storage 1 Type: CCPulse+ XML Data Files

Storage 2 Type: CCPulse+ 7.x Storage

Path To Storage 1: C:\G288_8110004_ENU\web_engagemen

Path To Storage 2: C:\Program Files (x86)\GCTI\CCPulse+\St

Storage 1 Contents:

WebEng_Chat_By_Agent
WebEng_Voice_By_Agent

Storage 2 Contents:

Sim_AllmediaVQ
Sim_KP_ Agent
Sim_Voice
Sim_VQ Evaluation
SimAll Media
SimChat Handling
SimE-Mail Handling
SimVoice Handling
SMS Queue Template
Stats on Agents
Survey Results
Task Monitoring Template
Tier
TierPFS
WebEng_Chat_By_Agent
WebEng_Voice_By_Agent

5. Repeat steps 3 and 4 for each template you would like to import into CCPulse+.
6. Click `Close`. The Import/Export Utility closes.

End