



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.

Developer's Guide

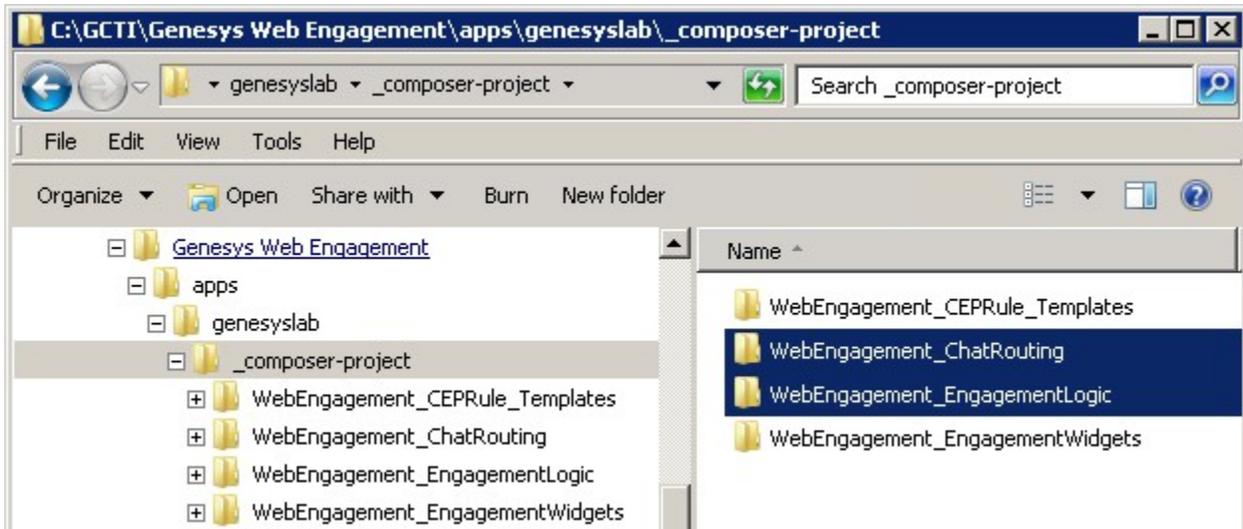
Customize the SCXML

Customize the SCXML

Contents

- **1 Customize the SCXML**
 - 1.1 Import the chat routing and engagement logic strategies
 - 1.2 Web Services
 - 1.3 Configure Authentication in the default SCXML strategy

When you create your application, Genesys Web Engagement also creates default chat routing and engagement logic strategies in the `\apps_composer-project\` folder.



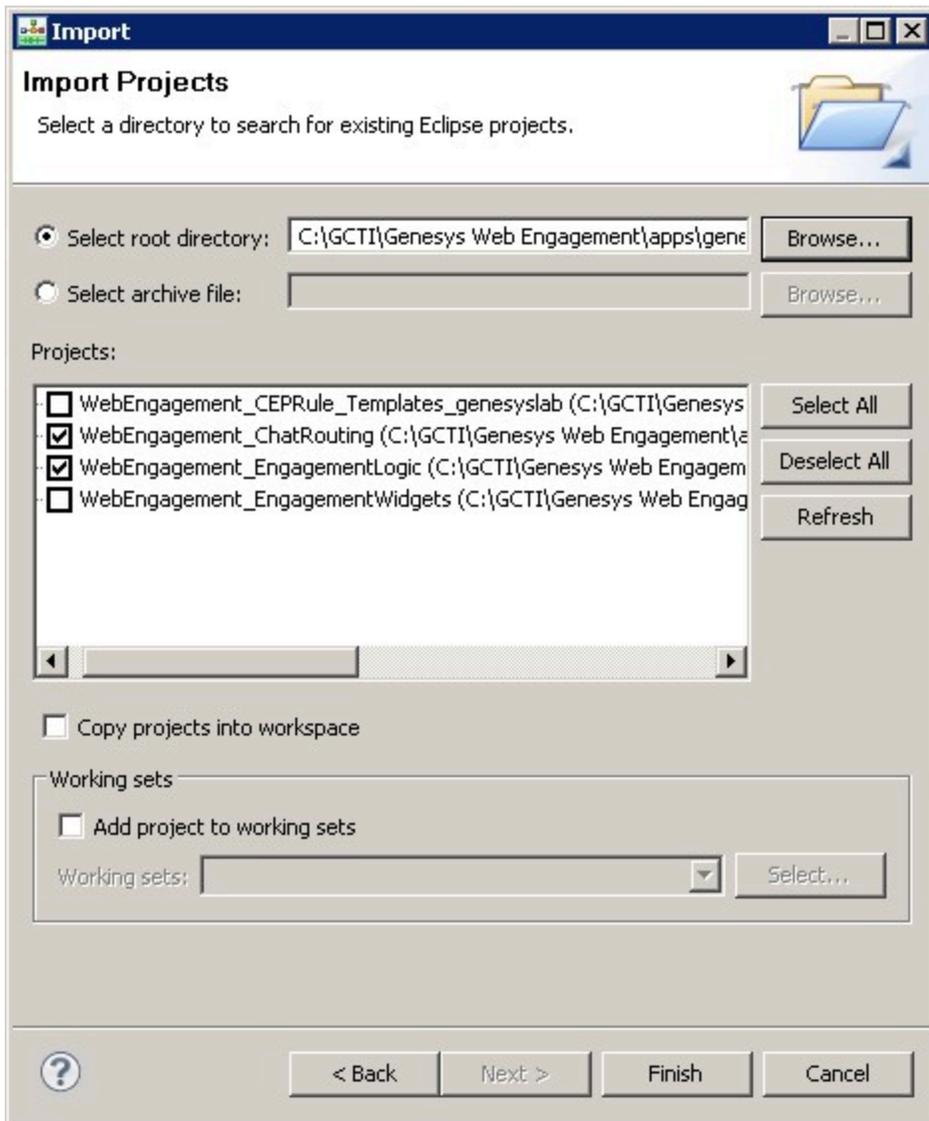
The default strategies for chat and engagement are in the `_composer-project` folder.

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.

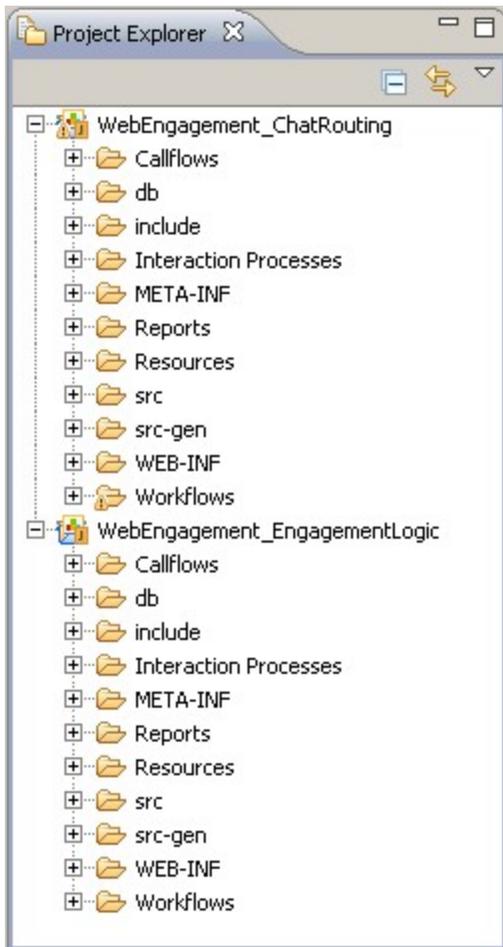
Import the chat routing and engagement logic strategies

1. Open Composer.
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:
 - Navigate to the `\apps_composer-project` folder and click OK. The list of projects available is added to the Projects list.
 - Select the `WebEngagement_ChatRouting` and `WebEngagement_EngagementLogic` projects.
 - Warning: Do **not** enable the option `Copy projects into workspace`.



Import the projects by clicking Finish.

- Click Finish to import the project. The WebEngagement_ChatRouting and WebEngagement_EngagementLogic project are added to the Project Explorer.



The chat routing and engagement logic projects are added to the Project Explorer.

End

Web Services

You can use web services in your SCXML in the following ways:

- Use a State block with `session:fetch`. For example:

```
<session:fetch srcexpr="BackendURL + '/data/gateway/engage'" method="'post'"
type="'application/json'" enctype="'application/json'">
  <content_expr="ixnProfile"/>
</session:fetch>
</onentry>
```

- Use a Web Request or Web Service block. In this case, Composer requires this logic to be hosted as a web application, which means the entire Composer project must be hosted outside of the Web Engagement application. With Composer, you can export the project as a web application in WAR

format.

You must set the Backend Server configuration option `wmsg.connector.scxml.appUrl` to the SCXML resources hosted outside of the Web Engagement application.

If you use queue-based routing, you must also set the `wmsg.connector.scxml.incomingInteractionQueue` option to your new queue name. Alternatively, you can go to the `Scripts\WebEngagement_EngagementLogic.queueBased.Incoming.Routing\Annex\Application` and change "url" to the new server.

Configure Authentication in the default SCXML strategy



Purpose: To add security credentials to the default SCXML strategy to support authentication for the REST API.

Prerequisites

- Your Web Engagement Backend Server supports authentication. See [Authentication](#) for details.
- Your strategy uses the REST API.

Start

1. Go to `\apps\<application_name>_composer-project\WebEngagement_EngagementLogic\src-gen` and open the `default.scxml` file in Composer, a text editor, or an XML editor.
2. Find the variables for user and password and set your authentication credentials. For example:

```
var user = 'user1';  
var password = 'password1';
```

3. Save your changes.

End