Composer Help

Creating VXML Applications

4/4/2026

Creating VXML Applications

Contents

When building any application in Composer, you first need to create a Project. A Project contains all the callflows, audio and grammar files, and server side logic for your application. By associating a routing strategy with a Project, you enable Composer to manage all the associated files and resources in the Project Explorer.

Cheat Sheet

Composer provides a cheat sheet to walk you through the steps for building a voice application.

- In the Welcome Screen (**Help** > **Welcome**), click the icon for Tutorials and select the Create a Voice Application tutorial. It will also describe the steps for how to make test calls and debug your application.
- If you are already inside the Workbench and Perspectives, access the same cheat sheet from the Menu bar at the top by selecting Help > Cheat Sheets, then Create Voice Application from the Building Voice Applications category.

Creating a New Project

You can follow the steps below to create a new Project:

1. For a Java Composer Project to be deployed on Tomcat, click the toolbar button to create a **Java Composer Project**. For a .NET Composer Project to be deployed on IIS, click the toolbar icon to create a .**NET Composer Project.**
2. In the Project dialog box, type a name for your Project.
3. If you want to save the Composer Project in your default workspace, select the **Use default location** check box. If not, clear the check box, click Browse, and navigate to the location where you wish to store the Composer Project.
4. Select the Project type:

    - **Integrated Voice and Route**. Select to create a Project that contains both callflows and workflows that interact with each other; for example a routing strategy that invokes a GVP voice application. For more information on both voice and routing applications, see What is GVP and How Do Voice Apps Work? and What Is a Routing Strategy, respectively.
    - **Voice**: Select to create a Project associated with the GVP 8.x. This type of Project may include callflows, and related server-side files. For more information on this type of Project, see topic, How Do Voice Applications Work.
    - **Route**: Select to create a Project associated with the Orchestration SCXML Engine/ Interpreter and Universal Routing Server. For more information on this type of Project, see topic, What Is a Routing Strategy.

5. Click Next.
6. If you want to use templates, expand the appropriate Project type category and select a template for your application. Templates are sample applications for different purposes. If you want to start from scratch, choose the Blank Project template and click Next.

7. Select the default locale and click Next.
8. Optional. If using the in a VoiceXML application, select the **Enable ICM** checkbox to enable integration. When checked, ICM variables will be visible in the Entry block. See the ICM Interaction Data block for more information.
9. Click **Finish**. Composer now creates your new Project. Your new Project folder and its subfolders appear in the Project Explorer.
10. To view/change settings not included under Preferences, right-click the Project and select **Properties**.

If you have never created a Composer Project, we recommend starting with Your First Application.

Creating a New Callflow

To add a new callflow diagram to an existing Composer Project:

1. Click the  button on the main toolbar to create a new callflow. Or use the keyboard shortcut: Ctrl+Alt+O.
2. In the wizard, select the tab for the type of the callflow. There are two main types of callflows in Composer represented by wizard tabs:

    - Main Callflow: Used for the main application where the call will land or be transferred to from another application.
    - Subcallflow: Used for modularizing your applications. It is useful for structuring large applications into manageable components.

Additionally you will benefit from the automated transaction reports associated with Subcallflows. Action Start and Action End VAR events are auto-generated for Entry and Exit blocks.

3. Select either Main Callflow or Subcallflow.
4. Select the type of diagram.
5. Click Next.
6. Select the Project.
7. Click Finish.
8. Create the callflow.

Validation

Composer can validate your diagram files and other source files for completeness and accuracy. For more information, see Validation.

Code Generation

The process of generating code creates a properly-formatted VoiceXML file from a callflow diagram built with Composer or a SCXML file from a workflow diagram. Static pages (pure VXML or SCXML code) are generated in the src-gen folder of the Composer Project. You can generate code in a couple of ways:

- Select Diagram > Generate Code.
- Click the Generate Code icon  on the upper-right of the Composer main window when the callflow/ workflow canvas is selected.

Note: If your project uses the Query Builder or Stored Procedure Helper-generated queries in DB Data blocks, the process of code generation will create one SQL file in the db folder for each such DB Data block. These SQL files will be used at runtime and should not be deleted.

Code Generation for Multiple Callflows

When using the Run as Callflow function, Composer automatically generates the VXML files from the diagram file that you want to run. When generating code, with the generate code function for a Java Composer Project that has multiple callflows, Composer attempts to generate the VXML for all the callflows before running (because the application might move between multiple callflows for subdialogs). However, if one of the callflows has an error, Composer provides the option to continue running the application anyway, because the erroneous callflow may be a callflow that's not used by the one being run (if there are two or more main callflows, for example). When this happens, the VXML files are basically out of sync with the diagram files and this may affect execution. Genesys recommends that you fix all errors before running the application.

Deploying/Testing Your Application

After you have saved your files and generated code for your application, test the application as follows:

1. Deploy the project for testing.

    - If deploying a Java Composer Project, Composer bundles Tomcat for running test applications, such as routing applications. If you configured the Tomcat settings prior to creating your Project, it will be auto-deployed on the Tomcat Server. You can double check this by clicking on the name of the project in the Project Explorer, then right-click and select Project Properties. Select the Tomcat deployment category and verify that the project is deployed. If not, click Deploy.
    - If deploying a .NET Composer Project, deploy your project on an IIS Server. Be sure you have configured the IIS settings. Click on the name of the project in the Project Explorer, then right-click and select Project Properties. Select the IIS deployment category and verify that the project is deployed. If not, click Deploy.

2. For Voice Projects, use Run mode to run the application by selecting Run > Run As > Run Callflow, or by right-clicking on the callflow file name in the Project Explorer and selecting Run As > Run Callflow. The code is generated in the src-gen folder and the debugger sends the call to your SIP Phone.

3. Accept the call and you will be connected to the application on GVP. The call traces will become visible in the Call Trace window, and you should hear the voice application run.