Composer Help

Deploying Composer Applications

4/4/2025

Deploying Composer Applications

Contents

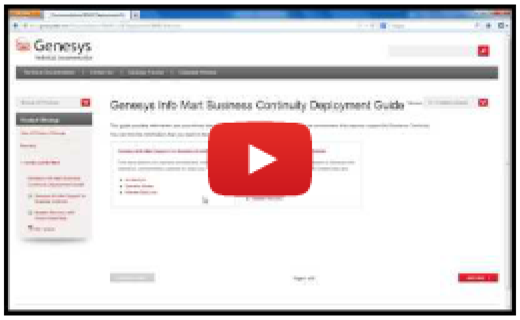
Consult the *Composer 8.1.4 Deployment Guide* for Composer product installation information. This topic describes deploying a Composer application to a web server.

Video Tutorial

Below is a video tutorial on exporting and deploying a Composer application to a web server.

Important

Important Note: While the interface for Composer in this video is from release 8.0.1, the steps are the basically the same for subsequent releases.



Deploying to Apache Tomcat Server for Testing

For testing purposes, Composer supports automated deployment of routing applications to the bundled Tomcat server or to a local IIS server. For more information, see Testing Your Application.

Migrating a Composer Application From Lab to Production

Automated deployment of a Composer Project application to application servers from within Composer (JBoss, Websphere, and IIS) is not directly supported. The Composer remote deployment option can be used for Java Composer Projects to deploy directly to remote machines. For more information, see the section on application server requirements in the *Composer 8.1.4 Deployment Guide*.

For the Web Service block, the Service End Point Variable Property can be used to externalize the Web Service URLs for production environments.

Deployment to a Web Application Server

Deployment to a web server depends on which type of Project you are working with:

- Java Composer Project
- .NET Composer Project

Once your application has been unit tested you will need to deploy it to a web server. The deployment process involves:

1. Exporting your Project
2. Transferring the files to your web/application server
3. Executing any necessary configuration steps required to make your application work.

For Composer-generated applications that use .NET resources, you can use any version of Microsoft IIS that is compatible with the Windows versions on which Composer is supported or any web server that supports Java Runtime Environment 1.7.0_0 or higher.

Combination Routing and Voice Projects

A single Composer Project can contain both routing and voice elements. If this is the case, the application will get deployed on a single application server (such has IIS or Tomcat), but must be provisioned in Genesys Administrator in two places:

- In GVP for the voice elements. See the chapter, Post Installation Activities on the GVP Hosts, Provisioning the Components section, in the *Genesys Voice Platform 8.1 Deployment Guide*.
- In URS for the routing elements. See the chapter, SCXML Strategy Support, in the *Universal Routing 8.1 Deployment Guide*.

Microsoft IIS Application Servers

Deploying an Composer application to a Microsoft IIS application server requires Administrative privileges when running the Microsoft Windows 7 and Microsoft Windows Server 2008, 32-bit operating systems.

Exporting a Java Composer Project for Deployment

Java Composer Projects can be exported to a location in the local machine and then manually deployed to any web application server that meets the following minimum pre-requisites:

- Must be J2EE 5 compliant.

- Must support the JSP 2.1/Servelet 2.5 specification (such as Tomcat andJBoss application servers).

Exporting a Java Composer Application to an Application Server

1. Select **File** > **Export**, expand the **Composer** folder, then select **Java Composer Project as WAR file** and click **Next.**
2. Select **Export Composer WAR file to a location** and click **Next**.
3. Select the Java Composer Project that you wish to export and specify the information below.
4. Enter **WAR Display Name**. This is the name that will be used for the application when it is deployed (for example, Tomcat Manager will show this name).
5. Enter **File system destination**. Enter the location on the local machine where the war file will be saved.
6. **Generate code for Composer diagrams**. Select the check box if you would like to auto-generate the code for all the callflows/workflows before exporting the `.war` file.
7. **Create Revision**. Select the checkbox and select **Major**, **Minor**, or **Micro**. This will be recorded in the Project history and can be reviewed later.
8. **Contributor** (Optional). Typically the username of the person exporting the Project.
9. **Comment**. Enter any comments associated with this operation. Typically these will indicate why a new version was exported.
10. Click **Finish**.

A file with the same name as the name of your Java Composer Project and file extension as `.war` will be created in the destination location. The `.war` file name is always set to the Project name. The display name is not the file name. It is the application display name used by an application server when the `.war` file is deployed in it. See the `web.xml` file in the `WEB-INF` folder in the generated `.war` file. It will contain the specified display name. When exporting a `.war` file, the display name is saved on a per-Project basis. Subsequently, when exporting the same Project, the saved name is pre-populated.

Once the `.war` file is exported, deploy the `.war` file in your Production server or JBOSS server manually. For example, typically, you will upload the file via Tomcat Manager or drop the file into the webapps folder of your Tomcat server. If your server is configured to auto-expand the files, you will see a folder created with the same name as your voice Project. If auto-expand is not configured you will have to stop and start your web server in order to expand the `.war` file.

Generating and Deploying a WAR file to a Remote Web Server

Use this option to export a `.war` file to a remote application server from within Composer. It removes the need to manually copy over the file to the application server to deploy it.

1. Select **File** > **Export**, expand the **Composer** folder, then select **Java Composer Project as WAR file** and click **Next.**
2. Select **Generate and Deploy a WAR File to a Web Server** and click **Next**.
3. Select the Java Composer Project that you wish to export and specify the information below. The

descriptions listed in the previous section applies here as well and are therefore not repeated.

4. **Web Container**: Select your target application server.
5. **Web Server installed location**: This is required for JBoss only. It is the location where the JBoss 711 application server was unzipped. It requires a file mapping to this location. Note: Only JBoss 711 version is supported for remote deployments.
6. **Generic Cargo configuration properties**:
    - **Host IP** – IP address of the host where the application server is currently deployed.
    - **Port No** – Webserver port.
    - **Username, Password** – Application server credentials required to deploy the application.
7. **Custom Cargo Configuration Properties**:
    - **Cargo** – Used by Composer for this remote deployment. Introduces additional capabilities, which can be specified via this option. Please refer to the Cargo documentation for details.
8. Click **Finish**.

Composer will generate a `.war` file and attempt to deploy to the specified remote application server. The information in the previous section also applies.

Deploying a NET Composer Project

.NET Composer Projects developed with Composer can be deployed on the IIS web application server.

- If you rename a .NET Composer Project, the new Project is not automatically deployed to IIS. The workaround is to undeploy the Project before renaming and then deploy manually after renaming.
- To deploy Composer .NET Projects to IIS, the IIS 6 Metabase Compatibility must be installed.

To deploy a voice application on IIS:

1. Generate the code for your project.
2. Right-click on your project and click **Export**.
3. In the Export dialog box, select **General** > **File System** and click **Next**.
4. Select all folders except `simulation`, `callflows`, and `debugging-results`.
5. In the To directory box, select the location in your file system where you want to export the application. Select the option for **Create only selected directories** then click **Finish** to export.

All your Project files should be exported to the location that you specified.

6. You can copy this folder to your final deployment machine.
7. Create a virtual directory for this application in IIS and point it to this folder:
    - In **Virtual Directory Alias**, specify the name that will be used to access this virtual directory from HTTP, then click **Next**.

- Browse to the folder that has your application's exported contents, then click **Next**.
- Give the following permissions: **Read**, **Run Scripts**.
- Configure the Mime types for the deployed .NET Composer Projects manually. The following mime types should be added: `.grx`
- Open IIS and select the website you want to use. Right-click it and select **New Virtual Directory**. A wizard dialog will be displayed. Click **Next** to start it.
- `ml` and `.vxml`

To add a mime-type, open Internet Services Manager and follow these steps:

- Right-click your website (such as Default Web Site) and select properties.
- Click the HTTP Headers tab.
- Click the **MIME Types** button to display the MIME Types dialog box.
- Add these MIME types:
  - `.grxml  application/srgs+xml`
  - `vxml   text/xml`
- Make sure that ASP.NET extensions are enabled in your IIS.
- Make sure that ASP.NET is enabled on your virtual directory and set to the correct version.
- Make sure that scripts have execute permissions on your virtual directory.

8. Right-click on the main vxml page in your `src-gen` folder and select **Browse**. If all settings are correct, a browser window will open and show you the VXML page. The address in the browser will be the URL at which your VXML application will be available.

Deploying a Routing Application

Deploying routing applications involves two main tasks.

1. Creating the appropriate objects in Configuration Server that are required by the Universal Routing platform. These objects are needed so that the platform understands how to direct interactions (voice or multimedia) as well as how to process them.
2. Generating SCXML pages accessible to the platform so they may be retrieved and processed by the platform.

You can handle both tasks in Composer using its integrated development environment.

- Publishing interaction process diagrams (IPDs) creates most necessary Configuration Server objects.
- Deploying Composer Projects to local application servers (Tomcat for Java Composer Projects and IIS for .NET Composer Projects) makes application SCXML pages available to the platform. However, Composer does not support deploying applications to a production environment. The steps documented below can be used to do that.

The term object is used (e.g., Interaction Queue object) when referencing Configuration Server objects. When blocks in Composer diagrams are referenced, the term block is used (e.g., Interaction Queue block). To deploy a routing application:

1. Deploy your Composer-generated SCXML pages to an application server. Note the URL of the starting SCXML page.
2. Using Genesys Administrator or Configuration Manager, log into the Configuration Database and connect to the Configuration Server that is being used for the environment in which you wish to deploy this application.
3. Create the Configuration objects listed below. All objects must be created under the appropriate Tenant that owns Configuration objects being referenced in your Workflow blocks like Queues, Standard Responses, and so on.
    a. One `Script` object for each Workflow block in your IPDs. The object type should be `EnhancedRouting`. This applies to workflows that process voice or multimedia interactions.
    b. One `Script` object for each Interaction Queue block in your IPDs. The object type should be Interaction Queue. This applies only to IPDs/ workflows that process multimedia interactions.
    c. One `Script` object for each View defined in Interaction Queue blocks in your IPDs. The object type should be Interaction Queue View. This applies only to IPDs/ workflows that process multimedia interactions.
    d. A media server Endpoint for each newly added Endpoint in any Media Server block in your IPDs. This applies only to IPDs/ workflows that process multimedia interactions.
    e. For any E-mail or SMS servers being referenced in Media Server blocks in your IPDs, configure the appropriate POP accounts to route multimedia interactions to the appropriate media server endpoints. This applies only to IPDs/ workflows that process multimedia interactions.
    f. Configure DNs to point to an `EnhancedRouting` object so that voice calls on those DNs invoke the application that is referenced in the Enhanced Routing object. This applies only to IPDs/ workflows that process voice interactions.

Detailed Steps for Creating Configuration Server Objects

The steps are listed below.

1. Common steps to create a`Script` object of a specific type:
    - Create a new `Script` object.
    - Ensure that the correct object type is set e.g. `InteractionQueue` or `EnhancedRouting`.
    - Set the object state to `Enabled`.
    - Check that the object is being created under the correct Tenant object.
2. Creating `Script` objects for Workflow blocks.
    - In Genesys Administrator, navigate to **Provisioning** > **Routing** / **eServices** > **Orchestration**.
    - Create a Script object of type Enhanced Routing. The name can be the name of your Workflow block. If another object already exists by this name, you can use a different name.

- Specify the URI. It should be the URL of the starting SCXML page of your application.
- Add a parameter: context_management_services_url. Its value should be in the format: [http:// http://]<UCS application host>:<Context Services port>.
- Additional parameters can be specified here. If the names match the names of any Project level variables, those variables will be initialized with values specified here.

3. Creating the Script object of type `InteractionQueue`.

- In Genesys Administrator, navigate to **Provisioning** > **Environment** > **Scripts**.
- Create a `Script` object of type Interaction Queue. The name can be the name of your Interaction Queue block. If another object already exists by this name, you can use a different name.
- In its Annex, create the following sections and keys:

Annex Section	**Property**	**Equivalent Composer Block**	**Value**	**Notes**
Namespace	Name	Name	<name of the queue>	
Namespace	Description	Queue Description	<descriptive text for the queue>	
Orchestration	Application	--	script:<name of the Enhanced Routing object to which interactions from this queue should be sent	Connects an interaction queue to an Enhanced Routing object. Equivalent to linking an Interaction Queue block to a Workflow block in an IPD.

4. Creating the `Script` object of type Interaction Queue View.

- In Genesys Administrator, navigate to **Provisioning** > **Environment** > **Scripts**.
- Create a `Script` object of type `InteractionQueueView.` The name can be the name of your view defined in an Interaction Queue block. If another object already exists by this name, you can use a different name.
- In its Annex, create the following sections and keys:

Annex Section	**Property**	**Equivalent Composer Block**	**Value**	**Notes**
View	Name	Name	<name of the queue>	
View	Queue	Parent Interaction Queue block	<name of the Interaction Queue object in Configuration Server	Connects the Interaction Queue View object with its parent Interaction Queue object

View	Description	Description	<descriptive text>	
View	Freeze Interval	Check Interval		
View	Condition	Condition		
View	Order	Order		
View	scheduling mode	Scheduling		
View	"Condition.<Name>"	Parameterized Conditions	<value>	
View	sql-hint	Database Hints		
View	segment-by	Configured Segments	Value will be "value of segment 1, value of segment 2, ..., value of segment n"	Segment names are not used
View	segment-check-interval	Segment Interval		
View	segment-total-limit	Segment Limit		

5. Creating Media Server Endpoints

   - In Genesys Administrator, navigate to the correct Application object representing your media server application of type EmailServer or SMSServer.
   - For EmailServer and SMSServer applications, Endpoints are created in the endpoints:<tenant dbid> section which is specific to a Tenant.
   - For each Endpoint, add a key in the above section with

     1. Key name = <Endpoint name>
     2. Value = <name of the Interaction Queue object to which interactions coming from this Endpoint will be submitted>

This defines an Endpoint and connects it to an interaction queue.

- Once an Endpoint is hooked up to an Interaction Queue object, all interactions coming in from that Endpoint are directed to the connected interaction queue. This object, in turn, is linked to an `EnhancedRouting` object from where the URL to the application is picked up. This completes the flow from the media server to the application SCXML pages.

6. Configuring POP accounts (EmailServer).

   - In Genesys Administrator, navigate to the correct application object representing your media server application of type EmailServer.
   - Accounts defined in EmailServer need to be configured to send e-mail interactions to specific Endpoints. For this, locate the pop-client<some number> section in the application’s options that represents a POP account.
   - In this section, set the value of the endpoint property to the name of the Endpoint to which e-mails should be redirected.

7. Configuring DNs. To connect DNs with SCXML applications for voice interactions, specify the following in the Annex of the DN:

Annex Section	Property	Equivalent Composer Block	Value	Notes
Orchestration	application	-	script:<name of the Enhanced Routing object to which interactions from this queue should be submitted>	Connects a DN to an Enhanced Routing object

a. For more details, see the chapter on configuring Orchestration Server in the Orchestration Server 8.1 Deployment Guide.

Deploying Applications That Use Context Services

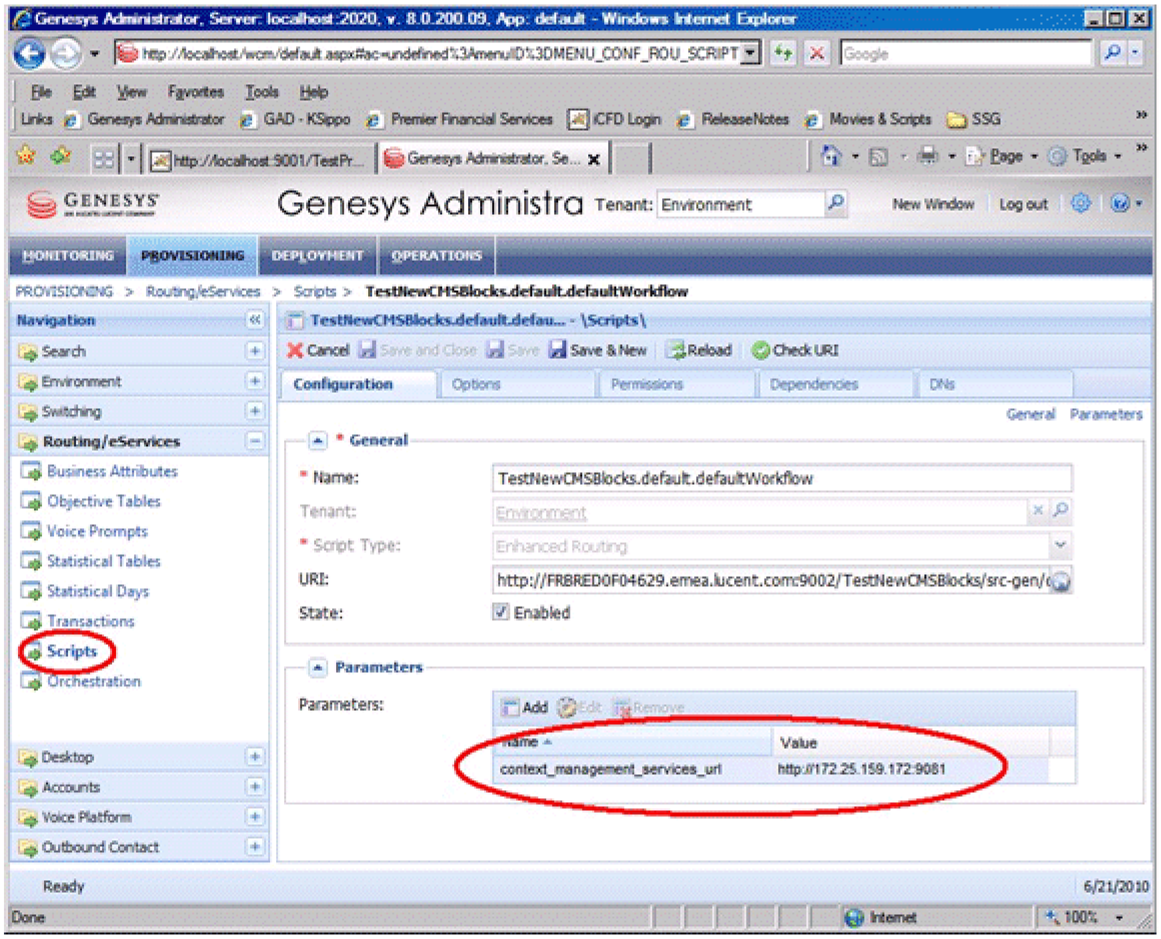
This topic discusses the following types of applications that use Context Services:

- SCXML Applications
- VXML Applications

SCXML Applications

When you publish an interaction processing diagram, Composer creates an enhanced Script record in the Configuration Database. This Script record has a `context_management_services_url` parameter, which is initialized with the UCS server parameters configured in Context Services Preferences. When the SCXML application is run, this parameter is read to enable Orchestration Server to connect to Universal Contact Server (UCS). If you want to point to another UCS, you must either:

- Update the UCS parameters in the Context Services Preferences and re-publish, or
- Manually update the `context_management_services_url` parameter using either Configuration Manager or Genesys Administrator. The example below shows the configuration in Genesys Administrator.



General

* Name:	TestNewCMSBlocks.default.defaultWorkflow
Tenant:	Environment
* Script Type:	Enhanced Routing
URI:	http://FRBRED0F04629.emea.lucent.com:9002/TestNewCMSBlocks/src-gen/
State:	☑ Enabled

Parameters

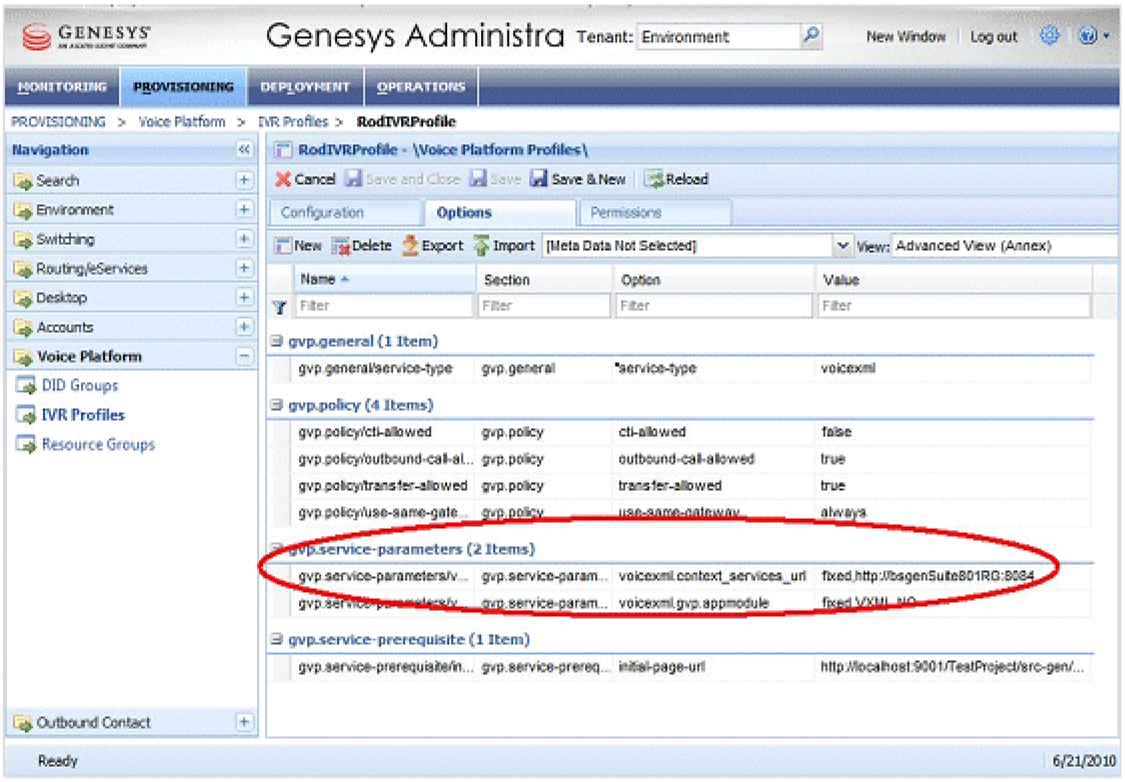
Parameters:

Name	Value
context_management_services_url	http://172.25.159.172:9081

MoreImages/contextServicesurl.gif

VXML Applications

When running a callflow from a RM Direct call, you must update the IVRProfile to define an additional `context_services_url` parameter whose value points to the Context Services (UCS) URL.



PROVISIONING > Voice Platform > IVR Profiles > RodIVRProfile

RodIVRProfile - \Voice Platform Profiles\

Configuration | Options | Permissions

[Meta Data Not Selected] View: Advanced View (Annex)

Name	Section	Option	Value
gvp.general (1 Item)			
gvp.general/service-type	gvp.general	*service-type	voicexml
gvp.policy (4 Items)			
gvp.policy/cti-allowed	gvp.policy	cti-allowed	false
gvp.policy/outbound-call-al...	gvp.policy	outbound-call-allowed	true
gvp.policy/transfer-allowed	gvp.policy	transfer-allowed	true
gvp.policy/use-same-gate...	gvp.policy	use-same-gateway	always
gvp.service-parameters (2 Items)			
gvp.service-parameters/v...	gvp.service-param...	voicexml.context_services_url	fixed,http://bsgenSuite801RG:8084
gvp.service-parameters/v...	gvp.service-param...	voicexml.gvp.appmodule	fixed,VXML_NG
gvp.service-prerequisite (1 Item)			
gvp.service-prerequisite/in...	gvp.service-prereq...	initial-page-url	http://localhost:9001/TestProject/src-gen/...

Ready 6/21/2010

Running a Callflow from a PlayApplication Workflow Block In this case, you must configure the `context_services_url` parameter in RM's default IVR Profile, which RM passes on to the VXML application. Configuration details are as follows:

1. In the Sip Switch/DN/VOIP Services/MSML_Service DN (if the msml-support option is true in Sip Server) or in the standard VoipService DN (if the `msml-support` option is false in SipServer): change the option contact from `sip:host_MCP:port_MCP` to `sip:host _RM:port_RM`
2. In the Tenant object, designate a default profile for `RM: gvp.general` section, option `default-application`=<name of some IVR Profile object under that tenant>, Default Application for instance.
3. In the IVR Profile/Default Application specified above, in the Annex, add the section `gvp.service-parameters`.
4. In the `gvp.service-parameters` section, add the option `msml.context_services_url=` fixed,http://demosrv8:908 (host:port of Context Management Server aka UCS' customer view port).
5. In the `gvp.service-parameters` section, add the option `voicexml.context_services_url=` fixed,http://demosrv8:9080 (host:port of Context Management Server aka UCS customer view port).

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**.

Note: 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.

Deploying Updates

This topic summarizes how to deploy updates to a Composer GVP voice application. For example, you may wish to deploy an updated version of a voice application with some new prompts and different DNIS numbers, but want to test it while the old one is still running. To deploy a new version of a Project without affecting the previously deployed one, try the following:

1. Export the existing Project (doesn’t have to be a new Project unless it is needed in cases where SCM tools are not used).
2. Rename the `.war` file before deploying to the application server. The deployed URL will depend on this war file name and therefore result in a different URL for the updated Project. You can control the display name of the application from the export wizard, however, this does not affect the URL.

Notes:

- The above procedure works on Tomcat, but has not been tested on other application servers.
- if the `.war` file is renamed before deploying to the application server, the application URL will reflect the .war file name instead of the Project name. This behavior may be application server-specific. The URL will look like:

```
[http:// http://]<ip_of_application_server>:<port>/<war_file_name>/src-
gen/<callflow_name>.vxml
```

- See the information on specifying the URL in Exporting a Java Composer Project above.
- Direct deployment to remote application servers is currently not supported in Composer.

Deploying Multiple Projects Using Bulk Manager

A new wizard, Composer Bulk Manager, is introduced in version 8.1.450.33 to manage bulk deployment of projects.

- Click the **Launch Bulk Manager** icon on the toolbar. The Composer Bulk Manager is displayed listing the Java and .NET projects available in the workspace, and their deployment status.

thumb|center|300px

- To deploy multiple projects, select the required projects and click the **Deploy Selected Projects** button. The selected projects are deployed and the status in the **Deployment** column is changed to *Deployed*.

- To undeploy multiple projects, select the required projects that have been deployed and click the **UnDeploy Selected Projects** button. The selected projects are undeployed and the status in the **Deployment** column is changed to *Not Deployed*.

Important

Both Java and .NET projects can be deployed in conjunction into their respective configured server locations using the Composer Bulk Manager wizard.