Composer Deployment Guide

Composer 8.1.4

12/30/2021

Table of Contents

Composer 8.1.4 Deployment Guide

This guide describes how to deploy Composer, an Integrated Development Environment used to develop applications for:

Voice applications for Genesys Voice Platform (GVP) 8.1+—a software suite, which unifies voice and web technologies to provide a complete solution for customer self-service or assisted service.

Routing applications for the Genesys Orchestration Platform 8.x, which includes:

- Universal Routing Server (URS)—which enables intelligent distribution of voice and multimedia interactions throughout the enterprise.
- Orchestration Server (ORS)—an open standards-based platform with an SCXML engine, which enables the customer service process.

ORS is responsible for executing orchestration logic (SCXML) that is provided by an application server (such as an application server hosting an SCXML-based routing application created in Composer). The responsibility of URS within the Orchestration Platform is to provide a necessary service to Orchestration Server to support Routing functions.

About This Document

This section includes:

Intended Audience

Contacting Customer Care

New in This Release

What is Composer?

This section introduces Composer:

Composer GUI

Debugging Applications

Other Composer Features

Language Support

Requirements and Support

This section includes information on:

Third Party and Other Software Requirements

Uninstalling and Reinstalling

This section describes:

Uninstalling and Reinstalling Composer

Application Server Requirements

Database, Speech Engine, SIP Phone, Web Browser Support

Before Installation

This section includes information on:

Installation Video

Installing as Eclipse Plugin

Update Site

Application Server Requirements

Links to Useful Docs

This section directs you to:

Orchestration Server Documentation

Orchestration Server Extensions

SCXML Technical Reference

Genesys Voice Platform Documentation

System-Level Guides

Installing Composer

This section includes information on:

Before Installing

Installing on Windows

Displaying the Interface

Windows 7 and Server 2008

Viewing a Sample Application

About Composer

Composer is an Integrated Development Environment, based on Eclipse, which Web Application developers can use to build VoiceXML, CCXML, GRXML, and SCXML applications.

Note: To familiarize yourself with basic Eclipse concepts, refer to the *Workbench User Guide* by selecting *Help > Help Content* s in the Composer main window and expanding the *Workbench User Guide* link.

Applications That Can be Developed

For the Genesys Voice Platform (GVP) 8.x Next Generation Interpreter (NGI), Composer provides the ability to develop:

- VoiceXML Applications with full support for Genesys Orchestration Extensions
- CCXML + VXML Applications requiring advanced call control features including conferencing.
- CTI + VXML Applications for Genesys Framework.

For the Orchestration Server 8.x SCXML Engine/Interpreter, Composer provides the ability to develop:

- SCXML-Based Routing Applications with support for SCXML for both voice and non-voice (multimedia) interactions.

Composer also provides Integrated CTI + VoiceXML applications for end-to-end treatment handling in conjunction with Genesys Voice Platform and Media Server.

For step-by-step instructions on using Composer, see the Composer 8.1.4 Help

This document is valid only for the 8.1.4 release of this product.

Intended Audience

This wiki is primarily intended for system integrators and administrators. It has been written with the assumption that you have a basic understanding of:

- Computer-telephony integration (CTI) concepts, processes, terminology, and applications
- Network design and operation
- Your own network configurations

You should also be familiar with the Genesys Framework 8.1 architecture.

While you can build applications by working solely with Composer’s blocks or Project templates, you

can also use State Chart Extensible Markup Language (SCXML) and VoiceXML when building applications.

Contacting Customer Care

If you have purchased support directly from Genesys, please contact Genesys Customer Care.

Before contacting Customer Care, please refer to the Genesys Care Program Guide for complete contact information and procedures.

New in This Release

See the *Composer 8.1.x Release Notes* and the *Composer 8.1.x Release Information* on the Composer product page.

Composer Interface

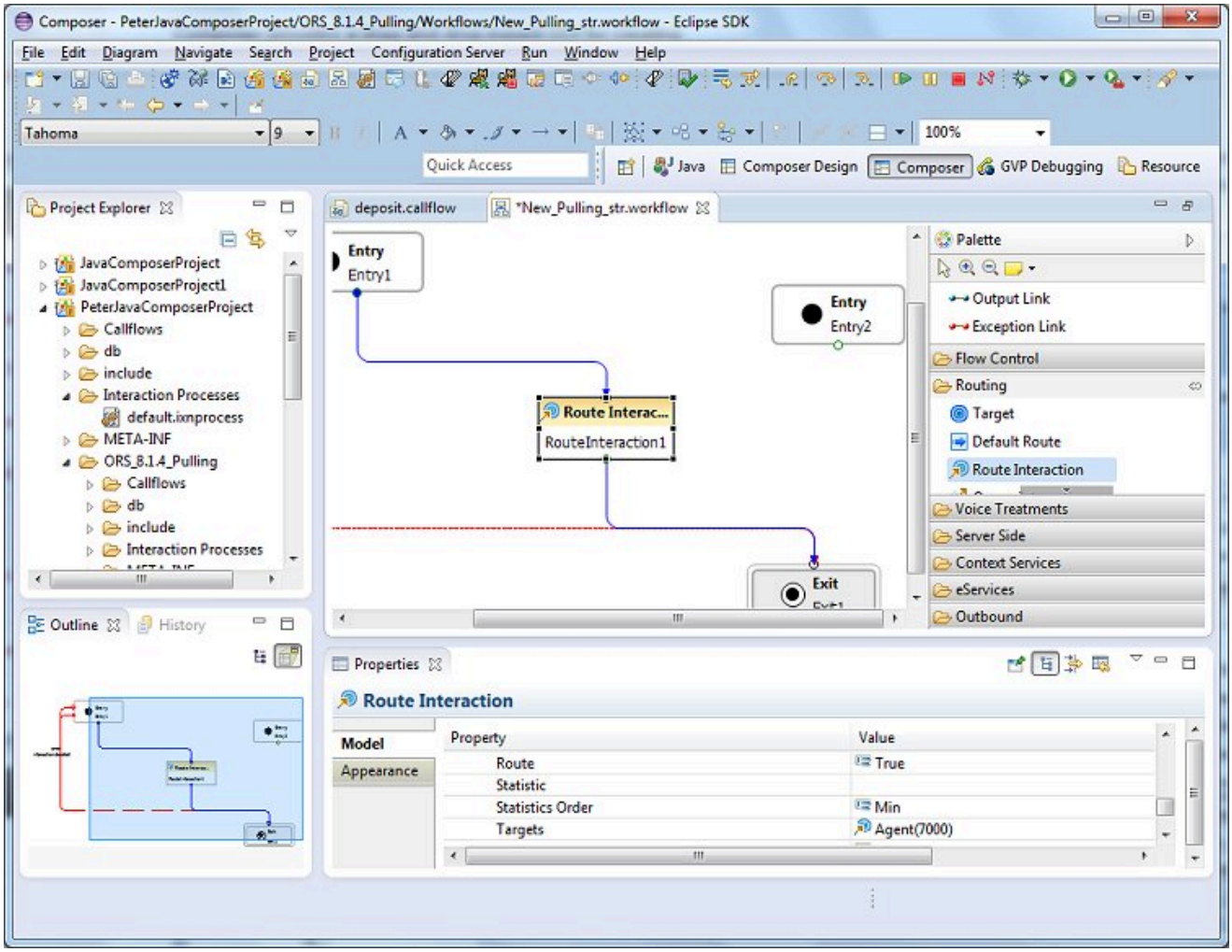
Note: This topic presents a brief overview of the Composer interface. For detailed information on using the interface, see the Composer Help.

Drag and Drop-Based GUI

Composer provides a drag and drop-based GUI for creating:

- VXML callflow diagrams (for voice applications)
- SCXML workflow diagrams and interaction process diagrams (for routing applications).

Technical and non-technical developers have the option of creating flow diagrams by placing and connecting blocks and configuring properties and/or by writing code. The figure below shows an example callflow in the center editing area in Composer perspective.



The interface elements in Composer perspective are as follows:

- A **Project Explorer** view on the upper left gives access to all the Project files.
- An **Outline** view of the entire callflow or workflow on the lower left is useful when working with complex diagrams.
- The **History** view on the lower left, which maintains previous versions of flows and application files, allowing you to revert to any previous version if needed.
- A center editing area (sometimes referred to as the **canvas** where you drag, drop, and connect blocks.
- A lower view for configuring block **Properties** (fields). Buttons in property rows display dialog boxes.
- A **Palette** of blocks grouped in categories on the upper right for creating flow diagrams.

A Composer perspective can also show various views in the lower pane depending on your actions or what you select from **Window** > **Show View** . For example, for voice applications, the lower pane can show the following views:

- **Properties**

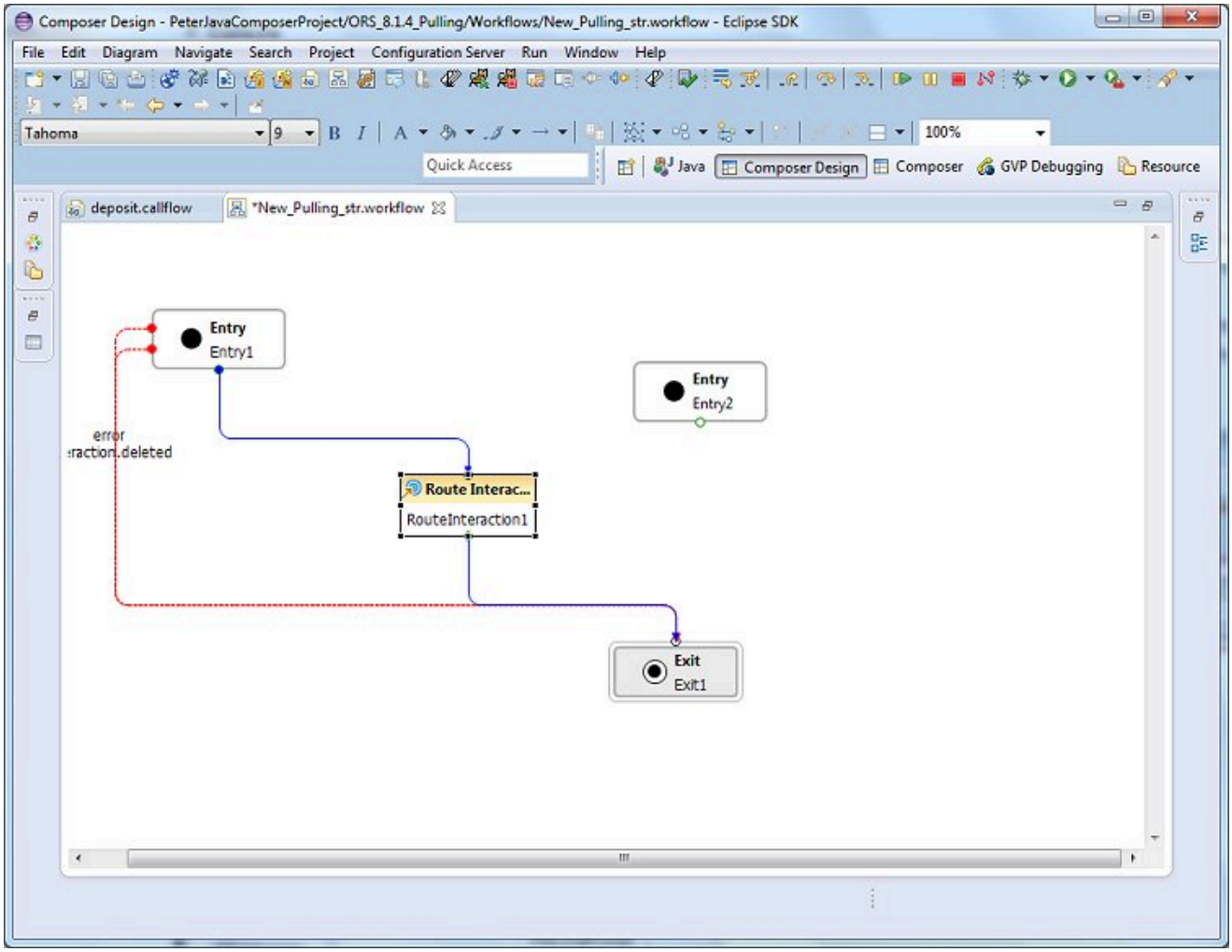
- **Prompts Manager**
- **Problems**
- **Console**
- **Call Trace**

Perspectives

When working in Composer, you have the option of working in different **perspectives** .

A perspective is an arrangement of different sections of the GUI in a manner that facilitates easy use of a particular feature, such as design or debugging. For example, the **GVP Debugging** perspective will show those sections that are useful when debugging a voice application: Call Trace, Console, Variables, Breakpoints, and so on.

The figure above shows **Composer** perspective. The figure below shows **Composer Design** perspective, which maximizes the design area. Having a larger design area is useful when creating complex flow diagrams. **Composer Design** perspective shows only the palette of blocks, the canvas area, and the *Properties* view, but can be customized to include other views that you select.



For routing applications, the lower pane can show the following views:

- **Properties**
- **Problems**
- **List Objects Manager**
- **Statistics Manager**

Available Perspectives

Composer includes the following perspectives for building applications:

- **Composer** , for both voice and routing applications, shows the Project Explorer, Outline view, canvas, palette, and can show the following tabs in the lower pane: Properties, Prompts Manager, Problems, Console, and Call Trace.
- **Composer Design** , for both voice and routing applications, can be used to simplify the workbench to

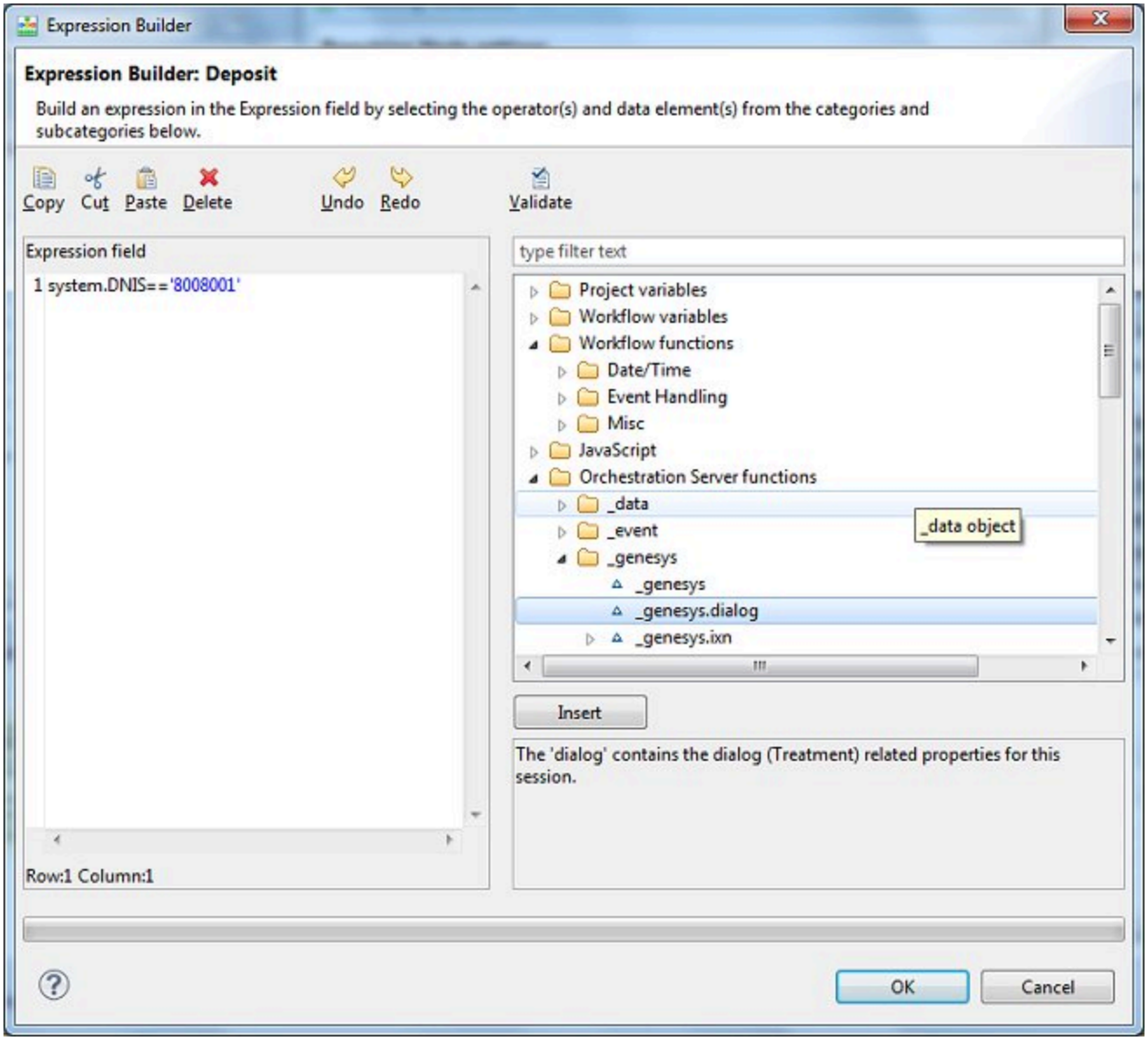
show only the palette of blocks, the canvas area, and the Properties tab.

- **GVP Debugger** , for debugging voice callflows that you build or import.
- **ORS Debugger** , for debugging routing workflows that you build or import.
- **Prompts Manager** , which provides the ability to quickly review all prompts in a Composer Project.

Expression Builder

Composer supplies Expression Builder to easily build expressions that can be used for branching and conditional routing decisions. You can also build expressions that use the Orchestration Server implementation of SCXML and Orchestation Server Extensions.

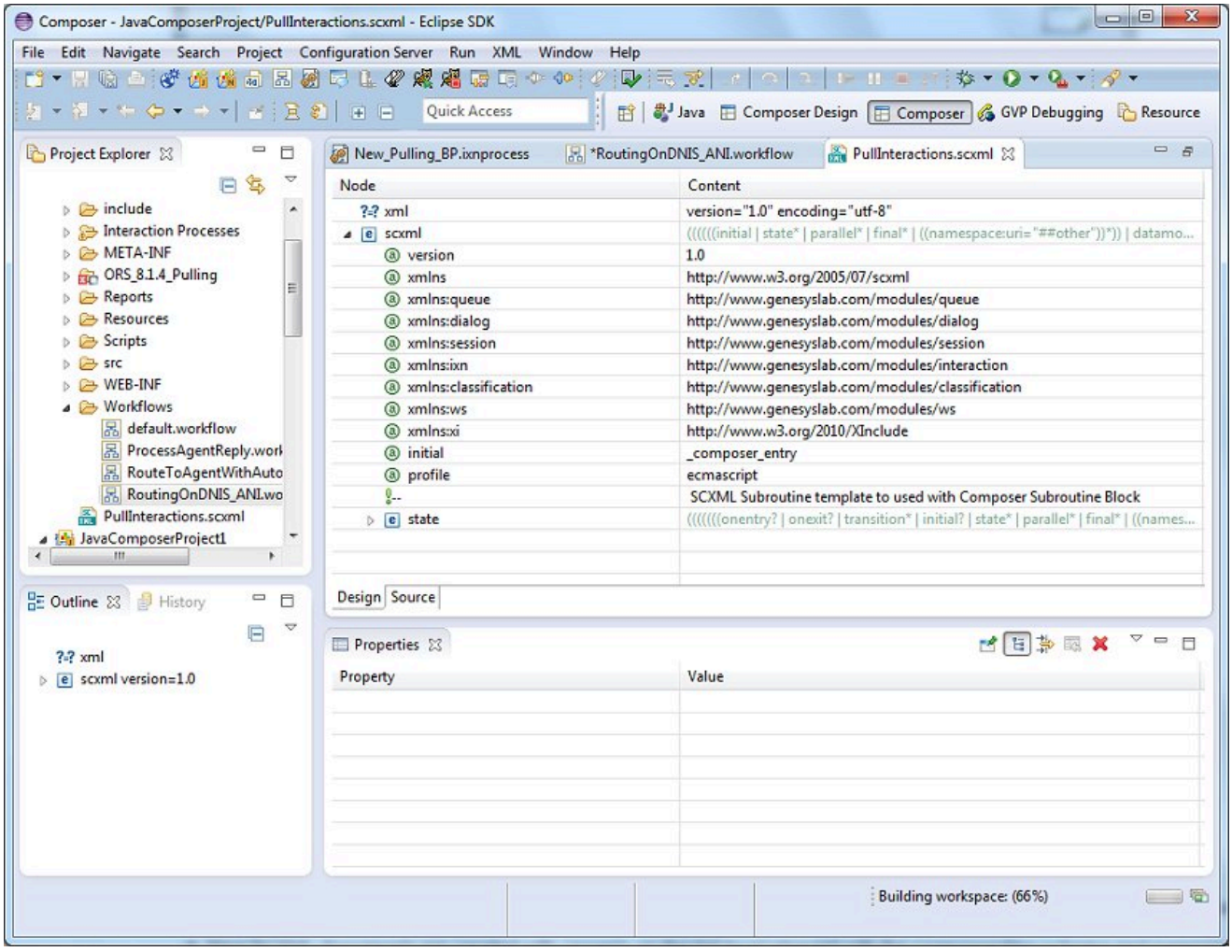
The figure below shows an example Orchestration Server function in Expression Builder.



Rich Editors

For those who prefer to write their own code, Composer provides a set of rich editors, supplying built-in error checking and tooltips, for SCXML, VXML, CCXML, and GRXML along with use case templates.

- The figure immediately below shows example SCXML code in the Design tab of the editor.



You can view and work directly with source code using standard Eclipse text editing features. Features include:

- Smart double-clicking behavior.
- Context-assisted help when typing tags. Also context-assisted help for attributes of a tag upon pressing Space inside a tag.
- New SCXML documents are created with <scxml> as the top level element with the corresponding schema and namespace specifications.
- Ability to edit tag attribute values from the **Properties** view.
- Basic editor actions are supported: Cut, Copy, Paste, Save, Save as, Undo, Redo, Search and Replace.
- Syntax highlighting.
- Show and hide Line numbers.
- Add/Remove Bookmark and To-Do markers.
- Task tag feature to auto scan To-Do comments in the code.

- Comparing and reverting to local file history.
- Spell checking by showing yellow squiggly line markers.
- Ability to see the outline structured view of the document in the **Outline** view.
- Validation shows errors in the *Problems* view. Validation happens based on the referenced schema.

Debugging VoiceXML Applications

Composer provides a real-time GVP Debugger with support for both **Run** and **Debug** modes.

- In the **Run** mode, call traces are provided and the application continues without any breakpoints.
- In the **Debug** mode, you can input breakpoints, single-step through the VoiceXML code, inspect and modify variable and property values, and execute any ECMAScript from the query console.

Integration with a SIP Phone is provided and click to dial feature is provided for making the test calls.

The Tomcat application server is bundled as part of the Composer and you can auto-deploy applications on Tomcat for testing.

Note: Composer 8.1 uses TCP to send SIP messages (previous releases used UDP). This is not a configurable option.

Debugging Routing SCXML Applications

Composer provides real-time debugging capabilities for Orchestration Server (ORS) routing applications. The Debugger is integrated within the workflow designer for making test calls, creating breakpoints, viewing call traces, stepping through an SCXML document/workflow, and debugging applications. Debugging can be started on an existing session or it can wait for the next session that runs the application at a given URL.

- Using a Run Configurations launch configuration, metrics (call traces) are provided and the application continues without any breakpoints. When the SCXML application executes, these metrics can describe, for example, state transitions, ECMAScript executions, and execution warnings or errors.
- Using a Debug Configurations launch configuration, you can input breakpoints, single-step through the code, inspect variable and property values, and execute any ECMAScript from the query console.

You can debug:

- A workflow built with Composer, or
- Any SCXML application or set of SCXML pages whether or not they were created with Composer.

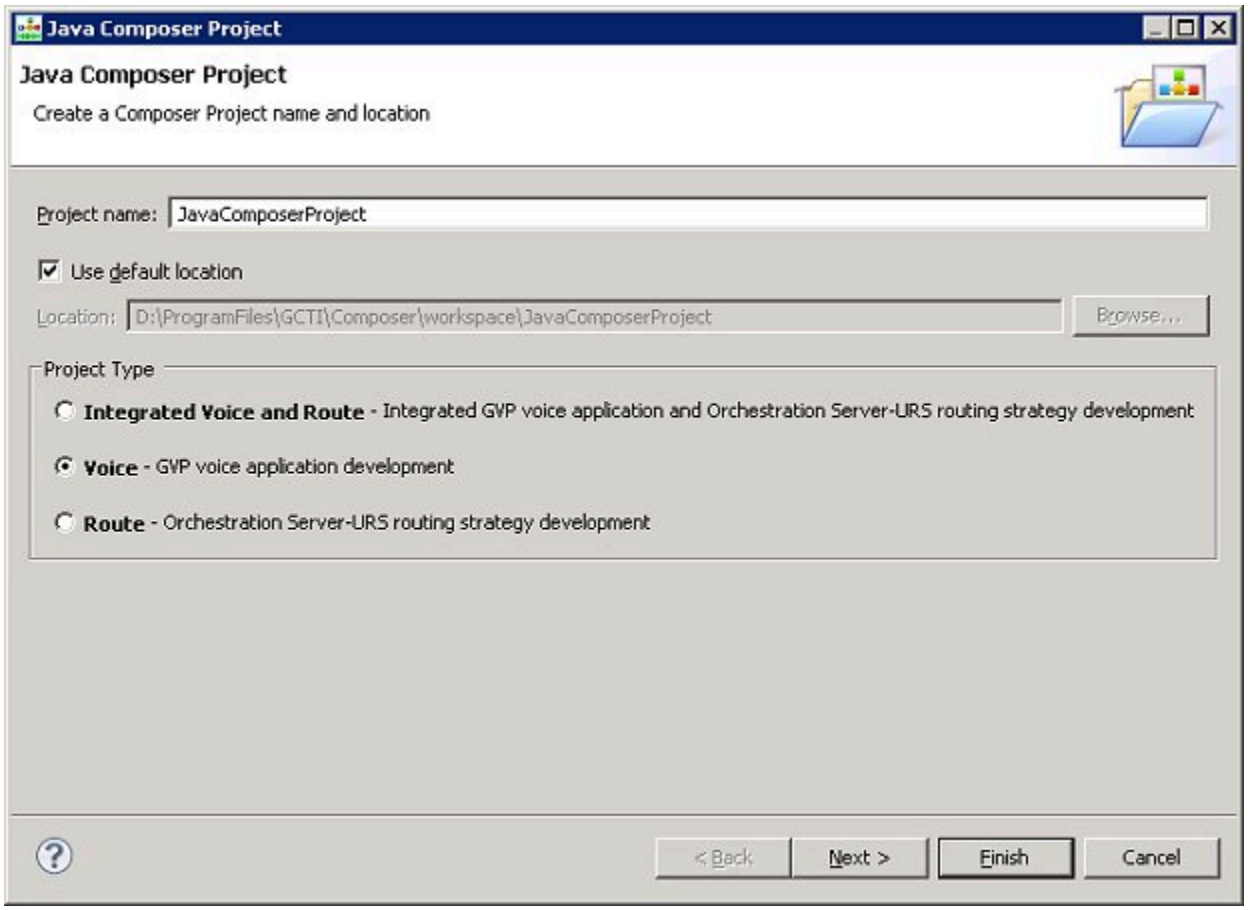
Other Composer Features

Some other Composer main features are summarized below. For details information on all Composer features, see the *Composer 8.1 Help* .

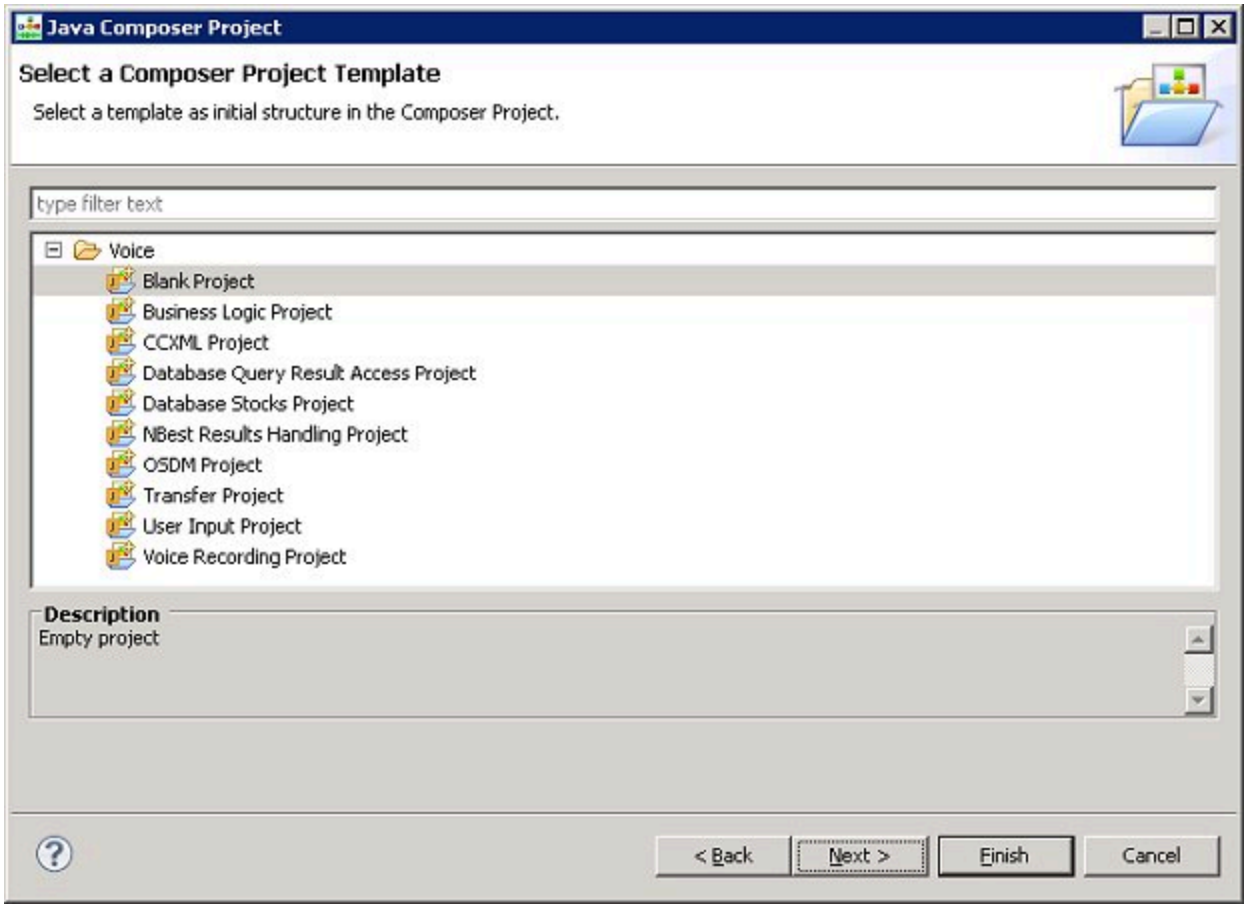
Project Templates

Out-of-the-box, reusable Project templates are provided. A Project wizard lets you select from three categories of templates:

1. **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.
2. **Voice** : Select to create a Project associated with the GVP 8.x. This type of Project may include callflows, and related server-side files.
3. **Route** : Select to create a Project associated with the Orchestration Server 8.1 SCXML Engine/ Interpreter.



Clicking *Next* brings up available templates for the selected category.



These can act as a starting point for new projects and visual flows and serve as guidelines and tutorials for routing and voice application developers. Composer also provides templates for its rich editors with the ability to create user-defined custom code snippet templates, which can be exported and imported to share across team members.

Code Generation

When generating code, Composer provides the ability to generate VXML pages to take advantage of the Platform optimizations. For SCXML routing strategies, Composer provides the ability to generate static SCXML pages for improved performance due to caching.

Deployment

Composer provides the ability to deploy Java Composer Projects and .NET Composer Projects. The deployment process involves exporting your project, transferring the files to your web/application server, and executing any necessary configuration steps required to make your application work. The Composer deployment process varies depending on the type of project being deployed (.NET Composer or Java Composer) and the associated application server. Future releases will provide the ability to deploy routing applications.

Project Management

Composer uses a Project to contain everything related to a single routing or voice application. A *Project Explorer* on the upper left of the Composer window contains all the Projects in your workspace. organize all the application elements.

Hiding Capabilities

Users may hide voice or routing capabilities through a Composer preference setting. This is useful for developers who are only using one of these Genesys platforms.

Builders/Managers

Composer contains several builders/managers, which are used for routing applications.

Statistics Builder/Manager

Use if you wish to use option of instructing Universal Routing Server to use the value of a statistic during target selection, such as *StatTimeInReadyState* . The statistic can be a URS Predefined statistic (as described in the *Universal Routing 8.1 Reference Manual*) or a statistic that you create yourself with Statistics Builder. Once you create a statistic, that statistic becomes available for selection in Composer's Target block.

List Objects Builder/Manager

A List object contains strings of any nature (for example, DNIS or ANI strings), which can be used in workflows. The strings can be as simple as 800 numbers or as complex as routing conditions. In Expression Builder, two URS Functions can be used to access List Objects: *_genesys.session.listLookupValue* and *_genesys.session.getListItemValue.*

Skill Expression Builder

Besides Expression Builder, Composer also has a Skill Expression Builder, which you can use for creating skill expressions used for routing decisions Opens from the *Targets* property in the routing Target block after selecting the *Skill* as the target type. Also opens from the Backend, Subdialog, Subroutine, Web Request, and Web Service blocks.

Customization Manager

The Customization Manager view helps you manage various aspects of your Composer installation that you have customized. You can manage any custom workflow and callflow diagram templates that you have created. You can also edit and delete custom templates, add new files, and save diagrams to disk.

Installation

This section describes the Composer software requirements, and installation and launching procedures.

Composer Installation Video

Below is a video tutorial on installing Composer 8.1.4 in Windows in an Eclipse environment. Depending on the flavor of Eclipse you have installed, your interface may appear slightly different than that shown in the video. The installation **Available Software** dialog box for release 8.1.430.06 adds the option for **Composer Language Locales** (not shown in this video). For more information, see Locales.

> **Important**
>
> Silent installation of Composer is not supported. Download the correct Eclipse for your computer's processor, i.e., download 64-bit Eclipse download or 32-bit Eclipse based on the target computer. Java Development Kit and Eclipse must match, i.e., both 32-bit or both 64-bit.

Getting Started After Installation

Installing Composer as an Eclipse Plugin

Composer 8.1.3/8.1.4+ is installed as plugins into an existing Eclipse environment as well as Oracle Java Development Kit 1.7.0_0 or higher. Starting with 8.1.410.14, the following Eclipse versions are supported:

- Eclipse Luna (4.4.2) and Mars (4.5). Other supported versions of Eclipse are Juno and Kepler.
- Oracle Java 7 is required. Oracle Java 8 compatible.

You may wish to consider "Eclipse IDE for Java EE Developers" if using Composer for Web and JavaScript-related development.

Note: Previous to 8.1.410.14, Indigo (3.7+) was supported for Composer 8.1.3 and some 8.1.4 versions. Not supported for Composer 8.1.410.14 and later. See Upgrading Eclipse.

Previous to 8.1.3, Composer installation consisted of:

1. The “basic” Eclipse application download.
2. Eclipse plugin dependencies.
3. The Genesys Composer plugins.
4. A bundled Tomcat for testing applications.

Starting with 8.1.3, the Composer installer supplies (3) and (4). The “basic” Eclipse (1) is up to you to download.

> ### Important
> Composer supports only Oracle JRE. All references to Java, JDK, and JRE in Composer documentation pertain to Oracle JRE.

Update Site

The Composer plugins are distributed as an *update site*, which is contained in *com.genesyslab.composer.updateSite.zip* . In Eclipse terminology, “update site” refers to a location from which Eclipse can pull plugins. The installer gives the option to pull the plugins from the update site zip file and install them into an existing Eclipse (1) environment.

If that option is enabled, then the Installer runs a script that automatically finds and downloads the plugin dependencies (2) and installs them along with the Composer plugins (3) and the bundled Tomcat (4). This is optional because you may not have downloaded a basic Eclipse yet, in which case you can install (2) and (3) later from within Eclipse using **Help** > **Install New Software** .

> **Important**
>
> Eclipse is not required first if you only want to extract the update site without installing Composer plugins into the Eclipse. This is possible, for example, if you just want the update site zip file to copy to multiple machines where the actual Composer installation will take place.

User-Facing Changes as a Result of Plug-in Installation

Change	Before 8.1.3	813+ Behavior
Change in file encoding used by Eclipse-based text editors to save files.	Composer based editors such as the VoiceXML Editor and SCXML Editor uses UTF-8 Encoding. Other Eclipse-based generic test editors uses default CP1252 encoding.	All the files and editors within the IDE will use UTF-8 encoding if the General > Workspace > Text File Encoding preference is default while launching the IDE. Otherwise, user-specified encoding takes precedence. However, Composer recommends using UTF-8 encoding.
The Composer splash screen is not displayed anymore.	Composer branded splash screen was displayed showing the Genesys logo, version, and so on.	Eclipse's default screen is displayed. No Genesys specific information shown.
Help > About	The Help About dialog is integrated into Eclipse and displays as part of the Eclipse workbench with information pulled out of the product definition.	The About Composer dialog exists is now a custom implementation. There is now a Help > About Composer in addition to a Help >> About.

Also see: Plugin Installation.

Operating Systems Supported

For information on supported operating systems, see the Composer section in the Genesys Supported Operating Environment Reference Guide. Also see the note on Windows 7 and Windows 8 Server under Installing Composer on Windows.

Application Server Requirements

Genesys does not certify Composer with specific Web application server vendors and versions. This applies to developing both:

- VXML applications and related resources that will be executed on the Genesys Voice platform and
- SCXML applications and related resources that will be executed on the Orchestration platform.

Java Composer Projects can be deployed to any Web application server that meets the following minimum pre-requisites:

- Application server should be J2EE 5 compliant.
- Support for the JSP 2.1/Servlet 2.5 specification.
- Oracle Java 7 or 8.

When considering a potential Web application server to use for a production deployment of Composer-generated applications, please refer to the vendor's documentation to ensure that the software meets these pre-requisites. In addition, you should run a few basic tests using a Composer sample application, to ensure that the application behaves as expected. For more details, please refer to the section Suggested Test Plan.

For developing SCXML applications and related resources that will be executed on the Orchestration Server platform, only specific Web application servers are supported. Please refer to the *Orchestration Server 8.1 Deployment Guide* for more information.

- Note: For more detailed information on deploying Composer applications to an application server, see Deploying Composer Applications.

Bundled Tomcat

Composer installs an embedded Tomcat web server for your use (code generation and testing). Genesys does *not* recommend that you use this bundled Tomcat web server for deploying and running your Composer-generated applications as part of a production setup. You may use a separate instance of Tomcat as a stand-alone web server, depending upon your needs and the recommendations of your IT department. Starting with 8.1.420.14, Composer supports Tomcat 7. Composer installation adds the role for `manager-gui` to Tomcat configuration for callflows and workflows.

For deployment of 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.

> **Important**
> Starting with release 8.1.450.20, the Tomcat version bundled with the Composer installation package is upgraded from 7.0.69 to 8.5.27.

Web Application Server Configuration

For Composer's server-side pages to work effectively on your Web application server, some configuration changes are required.

Tomcat

See the following sections ahead:

- Configuring Tomcat Settings
- Configuring Proxy Settings

Internet Information Services (IIS)

See the following sections ahead:

- Internet Information Services
- Configuring IIS/.NET Preferences
- Adding MIME Types for IIS
- Proxy Settings for .NET Composer Projects

Suggested Test Plan

After configuring your Web application server as described above, you should run some basic tests. Here is a suggested testing approach:

1. Create a new Project based on Project templates supplied with Composer. The New Project Creation wizard will guide you through the process and show a list of Project templates to choose from. Choose a template Project depending on the Project type and feature(s). Note: If you are using both the voice and route features, Genesys recommends that you test both features by running two tests.

Composer Feature	Not Using Databases	Using Databases
Voice	Business Logic Project	Database Project
Route	Routing Using Web Request Project	Database Query Result Access Project

2. These sample Projects may require configuration to be done, which will be documented in the workflow or callflow diagram(s) in the Project template. Projects that access databases contain a *readme.htm* file in the *doc* folder that provides instructions on how to set up the database as well as SQL scripts that may be needed to set up the required database structure and populate tables with sample data.

3. Validate the diagrams in your Project and verify there are no errors. Generate the code for these diagrams.
4. Export the Project for deployment. See the Deployment book in the Composer Help for the steps.
5. Consult the documentation for your application server on how to deploy applications on it. Some application servers may require custom steps.
6. Once deployed successfully, make a test call to invoke the application. Verify the application behaves as expected. If it does not, check the configuration and Troubleshooting book in the Composer Help for more information.

Database Support

For information on supported databases, see the Composer section in the Genesys Supported Operating Environment Reference Guide.

> **Important**
>
> When installing SQL Server, select SQL Server authentication (Composer does not support integrated Windows authentication).

Before you can use database blocks in a .NET Composer Project for accessing an Oracle database, you need to install and configure the Oracle client on the Composer machine. The Oracle client will be required on any deployment machines as well where the application will run. As a test, after installing the client software you should be able to connect to the Oracle database from SQLPlus. Once that works, database blocks in your .NET Composer Project should also be able to connect to your Oracle database. For configuring Oracle client, please contact your Oracle database administrator. Composer 8.1 has switched to an Oracle Provider from a Microsoft Provider for .NET Composer projects.

- At design time, both Java Composer Projects and .NET Composer Projects use bundled JDBC drivers to connect to Oracle, which is why your query will work in the Query Builder in both types of Composer projects. At runtime, Composer .NET projects use OLEDB database drivers, which are installed as part of Microsoft.NET Framework. These are required on each IIS where .NET Composer Projects will be run or deployed.
- Composer Java Projects use JDBC (Java Database Connectivity) drivers, which are bundled with Composer and are automatically installed with the software.
- Java Composer Projects continue to use JDBC, while .NET Composer Projects use Microsoft's OLEDB providers from the .NET Framework.

Speech Engines Supported

Composer supports all Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) engines that GVP 8.1 supports.

Web Browsers Supported

For information on web browser support, see the Composer section in the Genesys Supported Operating Environment Reference Guide.

Third Party Software Requirements

Composer requires the following third-party software on the computer on which Composer is installed:

1. **Eclipse must already be installed in your environment.** The only exception to this is listed above in the Update Site section.
   - Download the correct Eclipse for your computer's operating system, i.e., download 64-bit Eclipse or the 32-bit Eclipse based on your computer. Installation of Eclipse consists of extracting the downloaded zip file. The recommendation is to not extract the Eclipse zip file into a `Programs` folder. Composer installation allows you to specify the Eclipse path.
   - If installing Eclipse Juno EE 32-bit version,you must install the Oracle modeling plug-in, which Composer requires, but is not included with the Eclipse Juno EE 32-bit download. Note that many other dependencies are downloaded during the process that adds the Composer plugins to Eclipse. Genesys recommends using a dedicated instance of Eclipse for Composer.
2. **Oracle Java Development Kit 1.7.0_0 or higher must already be installed in your environment.**
   - Set the JAVA_HOME environmental variable to point to the JRE installation directory as described on the Oracle website. You must explicitly point to the Composer-supported version of Oracle JRE (1.7 or higher) and not use the system environment variable location. This is important if you have different versions of Java installed as the system environment variable may indicate another version, which may not be usable by the Composer version of Eclipse.
   - The above warning also applies to JDK installation. The JDK's JRE version could get lost if there are other JREs on a host. If you previously installed JDK separately (or did not enable the Public JRE option), Oracle JDK 1.7 installation by itself might require the Oracle JRE 1.7 installation to be explicitly called out. A symptom of this type of JRE confusion is Composer perspectives not being available after installation.
   - Oracle Java Development Kit and Eclipse must match, i.e., both 32-bit or both 64-bit.
3. **Microsoft .NET Framework** 2.0 and 3.5 (both are required if you intend to work with .NET Composer Projects).
4. **Web Service Enhancements** (WSE) 3.0 for Microsoft .NET. The WSE path must be specified in Composer's IIS/.NET preferences before Composer .NET Projects can work.

Note: Composer 8.1 does not support silent installation.

Genesys Software Prerequisites

To obtain the full functionality of Composer 8.1, the following Genesys products/software components are required:

- Orchestration Server 8.1.4 or later for developing and running SCXML session-based applications. To work with eServices, Orchestration Server 8.1.400.27+ is required.
- Orchestration Server 8.1.400.45+ is required if you wish to use the Associate New Interaction property for the Chat Transcript, Create Email, Create SMS, Email Forward, and Email Response blocks.
- Composer 8.1.4 and Interaction Server 8.5.1 are required for enhanced pulling of multimedia interactions. While other eServices components can be 8.1 Genesys recommends that other eServices be 8.5.0 or later.
- Genesys Voice Platform (GVP) Media Control Platform 8.1.6 or later for testing and running VXML applications.
- Genesys Configuration Server 7.5 or later for support of eServices capabilities.
- If you wish to use the Context Services capability of Universal Contact Server, you will need Universal Contact Server 8.1.000.10 or later.
- If you wish to use Outbound Campaigns, you will need Outbound Contact Server 8.1.100.09 or later.
- If you wish to use Business Rules, you will need Genesys Rules System 8.1.0 or later for business logic, which can be customized, and then invoked by VXML and SCXML applications.

Minimum System Requirements

Genesys recommends the following:

- Pentium 4 2GHz or comparable, 2 GB RAM or higher.

Minimum Screen Resolution

The minimum resolution for the Composer user interface is 1024x768 on a standard 4:3 aspect ratio monitor. The recommended resolution is 1280x1024. Lesser resolutions, such as 800x600, are not supported.

Before Installing

Before you install the Composer plugin, make sure that:

- You have met the **third party software requirements**.
- You have an account that has administrative privileges to install Composer.
- You have installed a supported SIP phone on your desktop (required if you want to conduct test calls). This can be done before or after installing Composer.
- You have installed Microsoft .NET Framework 2.0 *and* .NET Framework 3.5, for ASP.NET support in Composer.
- You have reviewed the Composer 8.1 Release Advisory.

- You have read the important information on User Account Control(UAC) (see note under Installing Composer on Windows).
- You have reviewed the chapter on configuring a security banner in the Genesys Security Deployment Guide as well as the Security Banner Configuration section below. If configuring a security banner, decide whether you want to configure it before or after installation of Composer.
- You are not installing through a Microsoft Remote Desktop Connection. You should perform the installation locally. Genesys does not recommend installation of its components through Remote Desktop Connection.
- Composer 8.1 does not support silent installation.

Security Banner Configuration

Decide whether you wish to configure a security banner.

Composer installation gives the option of configuring a security banner that displays when users connect to Configuration Server. In the case of Composer, the security banner appears in a separate window that is displayed when you connect to Configuration Server. The content of this window is defined by the system administrator, and can include such items as Terms of Use of the application or some kind of disclaimer.

While Composer supports the basic operation of specifying and displaying a security banner as described in the Genesys Security Deployment Guide, be aware of the following limitations for Composer:

- The banner only displays when you connect to Configuration Server after launching Composer.
- The following options are not supported:
  - List of security banner URLs.
  - **Error URL** if banner cannot be retrieved.
  - **ShowUpTimeout** , which displays a *Loading* message if loading takes more than this timeout.
  - **NoCompleteTimeout** , which is the timeout for retrieving security banner.
  - **AckMandatory** , which controls if user must acknowledge the security banner in order to proceed.
  - Height, width and title of dialog.
  - **AckMode** , which controls how often the banner is displayed. Note: A Java equivalent of unilogin library is not available.
- The security banner configuration option *Until each user chooses to turn it off once for application type* is not supported. It is handled the same as the option *Until each user chooses to turn it off* .

Under some circumstances, Composer may not try to display a banner page although the loading of a previous page in the list failed. This behavior is likely to happen if the embedded browser component returns that loading is finished (although the loading failed) before the loading timeout elapses.

Pending Operations Message

When installing or un-installing Composer in a Windows XP Professional or Windows Server 2003 Standard Edition environment, the prompt to reboot appears. After the reboot, the following message appears:

```
There are some pending operations and the system needs a reboot.
The target computer does not meet some mandatory requirements.
```

Subsequent reboots result in the same behavior and you temporarily cannot install Composer on this machine. This is a result of pending reboots from other installations. If this situation occurs, follow the procedure below.

Registry Edit for Pending Operations Message

Use the information below if you receive the above message after un-installing or installing Composer:

1. Open the Registry editor.
2. Remove the following key: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\PendingFileRenameOperations**
3. Install Composer as described below.

Installing Composer on Windows

At a high level, the installation process for is as follows:

1. Install Eclipse.
2. Configure update site for the specific flavor of Eclipse.
3. Install JDK and select the public JRE option.
4. Shutdown Eclipse.
5. Install Composer and select the **Composer Plugins** option.

A video of the Composer installation procedure is available at the beginning of this page.

Notes:

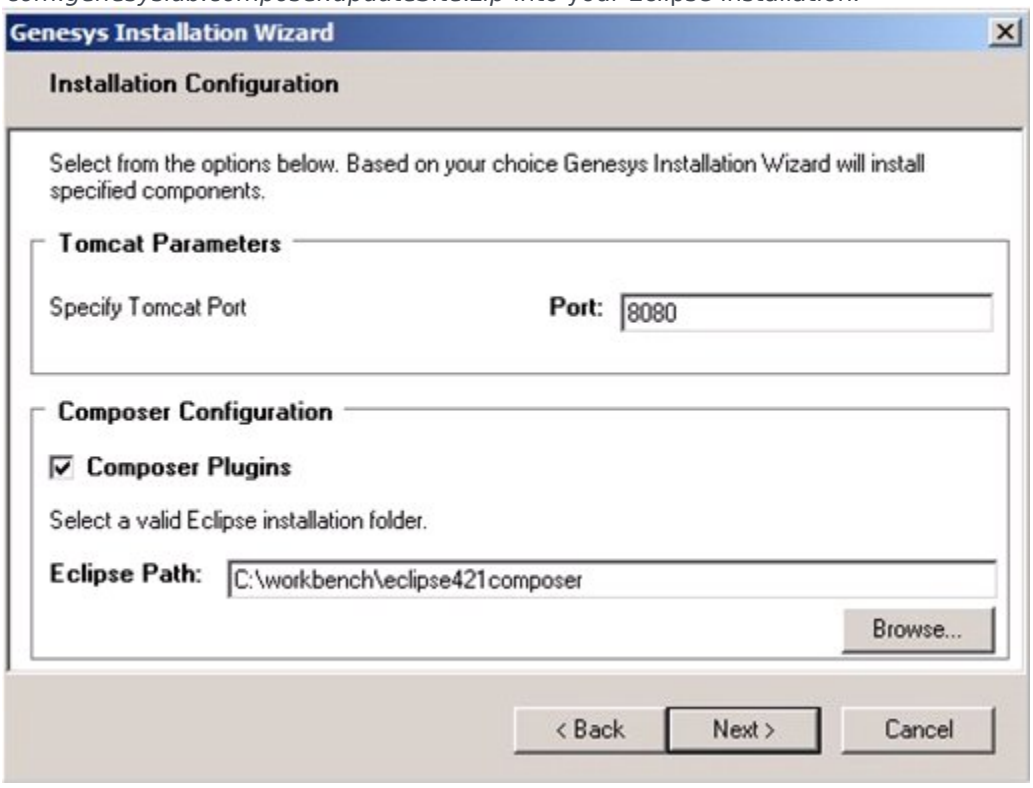
- On Windows 7 and Windows Server 2008, UAC (User Account Control) can affect the installation process and subsequent addition of Composer plugins to Eclipse. If UAC is disabled, installing Eclipse in a Program Files location (any location that is under UAC control) will work. However, if UAC is enabled, then Eclipse should be installed in a location that is not under UAC control e.g., C:\GCTI\Composer.
- While installing Composer 8.1.4, there is a step to download dependencies from Eclipse update sites. If your Internet connection is lost during this process, or some other error occurs, the Installer may not detect it and may incorrectly report success. Error messages will appear in the *scripts\composer-plugin-install.log* file in the Composer installation directory, (Example: *C:\Program Files\GCTI\Composer\scripts\composer-plugin-install.log*). Workaround: Instead of letting the Installer add Composer plugins directly to Eclipse, add Composer plugins using Eclipse's "Install New Software" feature (see Manual Plugin Installation below).

Prerequisites

See Third Party Software Requirements.To uninstall/reinstall a later version, see Uninstalling and Reinstalling.

Installation Steps

1. Run **setup.exe** from the Composer installation package (or where the latest version is located).
2. If you want to configure a security banner, see Security Banner Configuration section above.
3. Enter **Destination**. The destination is also where com.genesyslab.composer.updateSite.zip will be located.
4. When you reach the screen shown below:
   - Check the **Composer Plugins** checkbox and browse to the existing Eclipse location (the folder where `eclipse.exe` is located). This option automatically installs *com.genesyslab.composer.updateSite.zip* into your Eclipse installation.



   - You also have the option to leave the **Composer Plugins** box unchecked and manually install the Composer plugin. A common case for leaving Composer Plugins unchecked would be if Eclipse is not yet installed or you do not have internet access so that dependencies cannot be downloaded during the installation process. See Manual Plugin Installation section below.

The next step assume you have checked the **Composer Plugins** box.

5. Specify the Tomcat port. For testing purposes, Composer supports automated deployment of routing applications to the bundled Tomcat server or to a local IIS server. Enter a port number to be used for Tomcat.

> **Important**
>
> If there is already a version of Tomcat installed, the port should be changed from 8080, which is the default, to another port (for example, 8090) to avoid port conflicts.

6. Click **Next** and **Finish** on the installation wizard. The installation may take some time because other plugin dependencies are also downloaded and installed.
7. To open Composer, open Eclipse. You may have a desktop shortcut or you may need to search for Eclipse in the directory where you extracted the Eclipse zip. If a Welcome screen appears, close it.
8. Select the Composer perspective by clicking the **Composer** button in the upper right of the screen. Or select **Window** > **Perspective** > **Open Perspective** > **Other**. All perspectives (views) available in Eclipse are listed, including those not used by Composer. Select **Composer** perspective.

Next Steps After Automatic Plugin Installation

- See Post Installation Configuration.
- See Composer Interface.

Manual Plugin Installation

If you left **Composer Plugins** unchecked, you can later manually install the Composer plugins if:

- You have an internet connection
- You have installed the required versions of Eclipse and Java Development in your environment.
- You have enabled the Public JRE option, which causes the JDK installer to install both JDK and JRE. Composer requires both components.

The Composer installation will have created a directory, e.g., *C:\Program Files\GCTI\Composer 8.1* , in which there is a zipped update site called *com.genesyslab.composer.updateSite.zip* .

1. In a running Eclipse instance, go to **Help** > **Install New Software** .
2. If installing Composer in Eclipse Indigo 3.7 (not supported starting with 8.1.410.14 and later), the following extra steps are required.
   - On the Eclipse **Install' *dialog box, click*** *Add* .
   - Opposite **Name** , enter **GMF Tooling** .
   - Opposite Location, enter the URL for GMF Toolng: *http://www.eclipse.org/modeling/gmp/downloads/?project=gmf-tooling*
   - Click **OK.**

3. Click **Add.**
4. Type something in the *Name* field, such as *Composer* .
5. Click **Archive** .
6. Browse to the *com.genesyslab.composer.updateSite.zip* file in the *Destination* folder you previously specified in the **Destination** field. The destination is also where com.genesyslab.composer.updateSite.zip will be located.
7. Click **OK** .
8. Check the three Composer Features (*Common*' , **Voice, Route**).
9. Click **Finish** , and click through the license pages. The installation will now begin. Again, this may take some time because other plugin dependencies are downloaded and installed.
10. After installation is complete, the Installer will prompt you to restart Eclipse. **Click OK** .

Windows 7 and Server 2008 32 Bit

The procedure below also applies to 64-bit Windows 7 Ultimate in 32-bit compatibility mode.

1. Run / launch Eclipse as Administrator.
2. DO NOT keep your workspace under the *Program Files* location as prompted by Composer. Instead specify your workspace location outside of the *Program Files* location e.g., *C:\My Composer 8.1.4 Applications* or *C:\Users\<your user>\MyWorkspaceFolder*

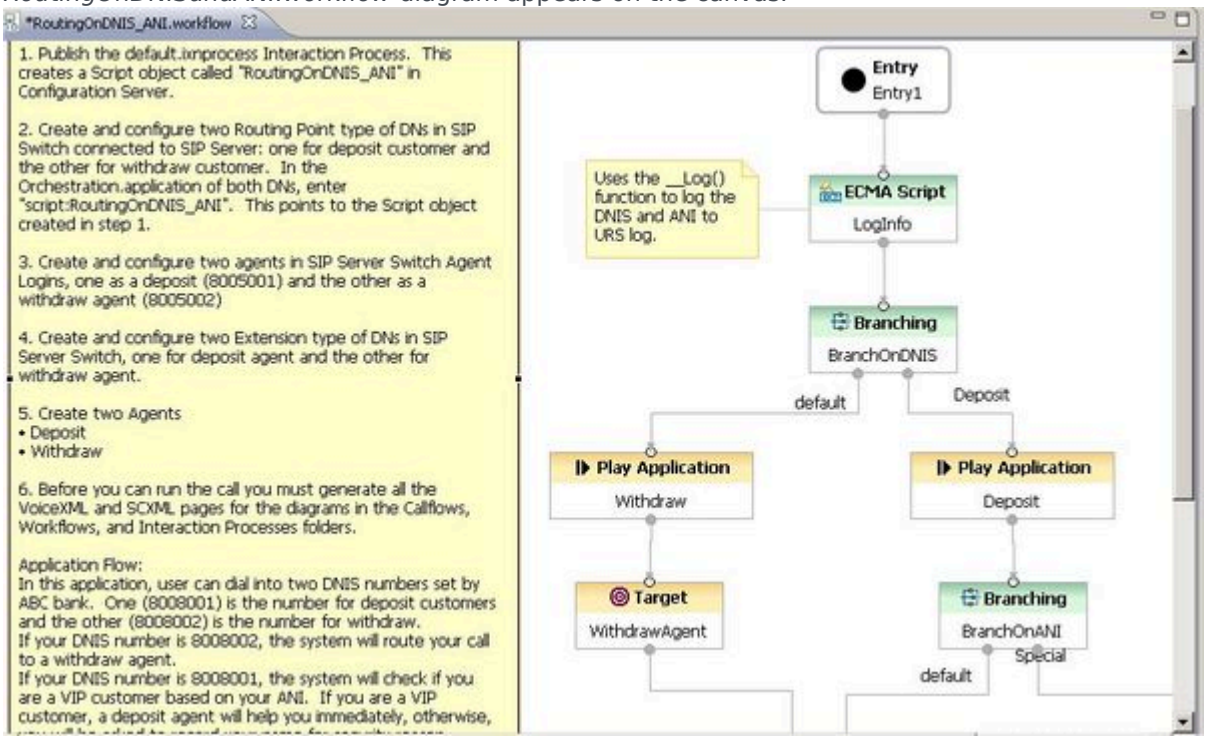
Viewing a Sample Application

Composer provides a set of predefined Project templates containing sample applications. If using Composer for the first time, before creating your own Project, you may find it helpful to load one or more of the sample Projects. The procedures below lead you through the process of loading a sample applications.

Sample Integrated Voice and Route Application

1. By default, when you enter Composer for the first time, you will be taken inside the Composer perspective.
2. On the toolbar, click the button to create a Java Composer Project.
3. In the **Java Composer Project** dialog box, name your Project and indicate whether you want to use the default location.
4. Select the Project type: **Integrated Voice and Route' ,** *Voice* , or **Route** . Your selection will determine which Project templates are shown in the next dialog box. For this example, select *Integrated Voice and Route* .

5. Click **Next** . The **Select a Composer Project Template** dialog box opens.
6. Select the **Routing Based on DNIS and ANI** Project template and click **Finish** . A commented *RoutingOnDNISandANI.workflow* diagram appears on the canvas.



Double-clicking a block opens its **Properties** view so you can view the fields.

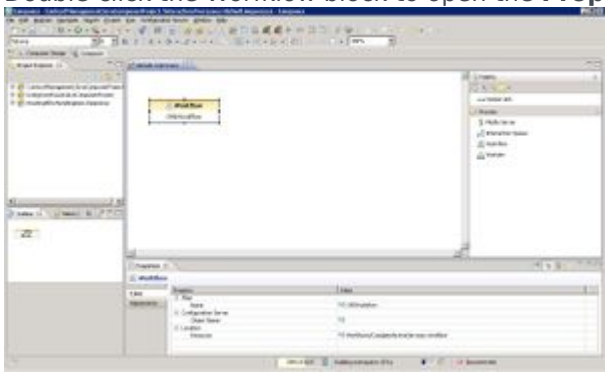
The next procedure differs from the above procedure in that it demonstrates how to view a workflow contained within an interaction process diagram.

Viewing a Routing Workflow Inside an Interaction Process Diagram

By default, when you enter Composer for the first time, you will be taken inside the Composer perspective.

1. On the toolbar, click the button to create a Java Composer Project.
2. In the **Java Composer Project** dialog box, name your Project and indicate whether you want to use the default location.
3. Select the Project type: **Integrated Voice and Route** , **Voice** , or **Route** . For this example, select **Route** .
4. Click **Next** . The **Select a Composer Project Template** dialog box opens.

5. Select **Context Services Service Project** click **Finish** . This automatically creates an interaction process diagram for voice interactions with a single Workflow block in the *default.ixnprocess* tab.
6. Double-click the Worfklow block to open the **Properties** view in the tab underneath.



7. In the **Properties** view, note that the **Resource** property indicates that the name of the workflow is *CompleteActiveServices.workflow* . To view this workflow, expand the Project in the **Package Explorer** on the left.
8. Expand the **Workflows** folder.
9. Double-click *CompleteActiveServices.workflow* . A commented workflow appears.
10. View the properties for each block by double-clicking a block.

Use the above method to review the various routing Project templates.

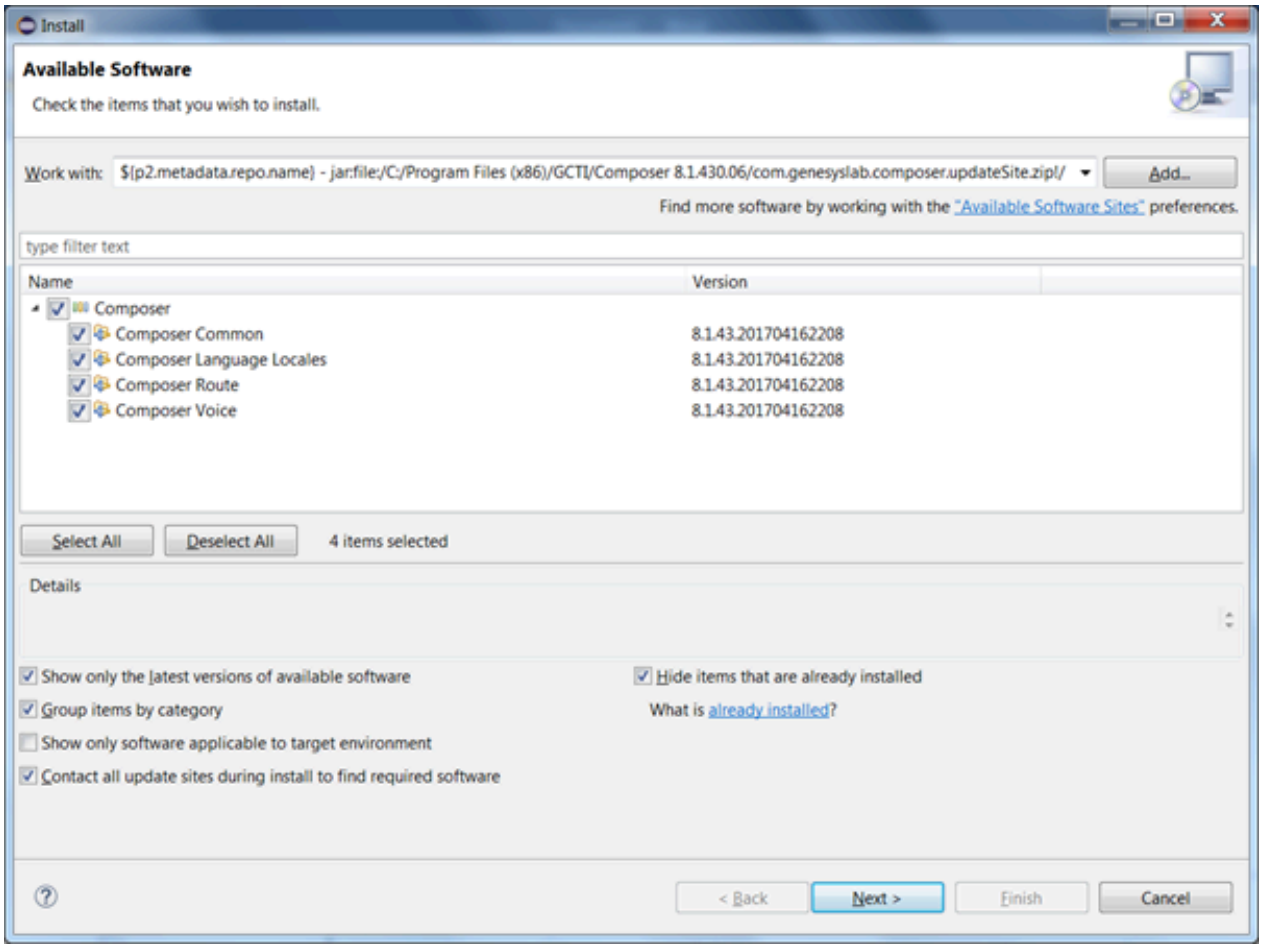
Language Locales

Composer versions 8.1.430.06 and 8.1.450.33 support French Canadian localization.

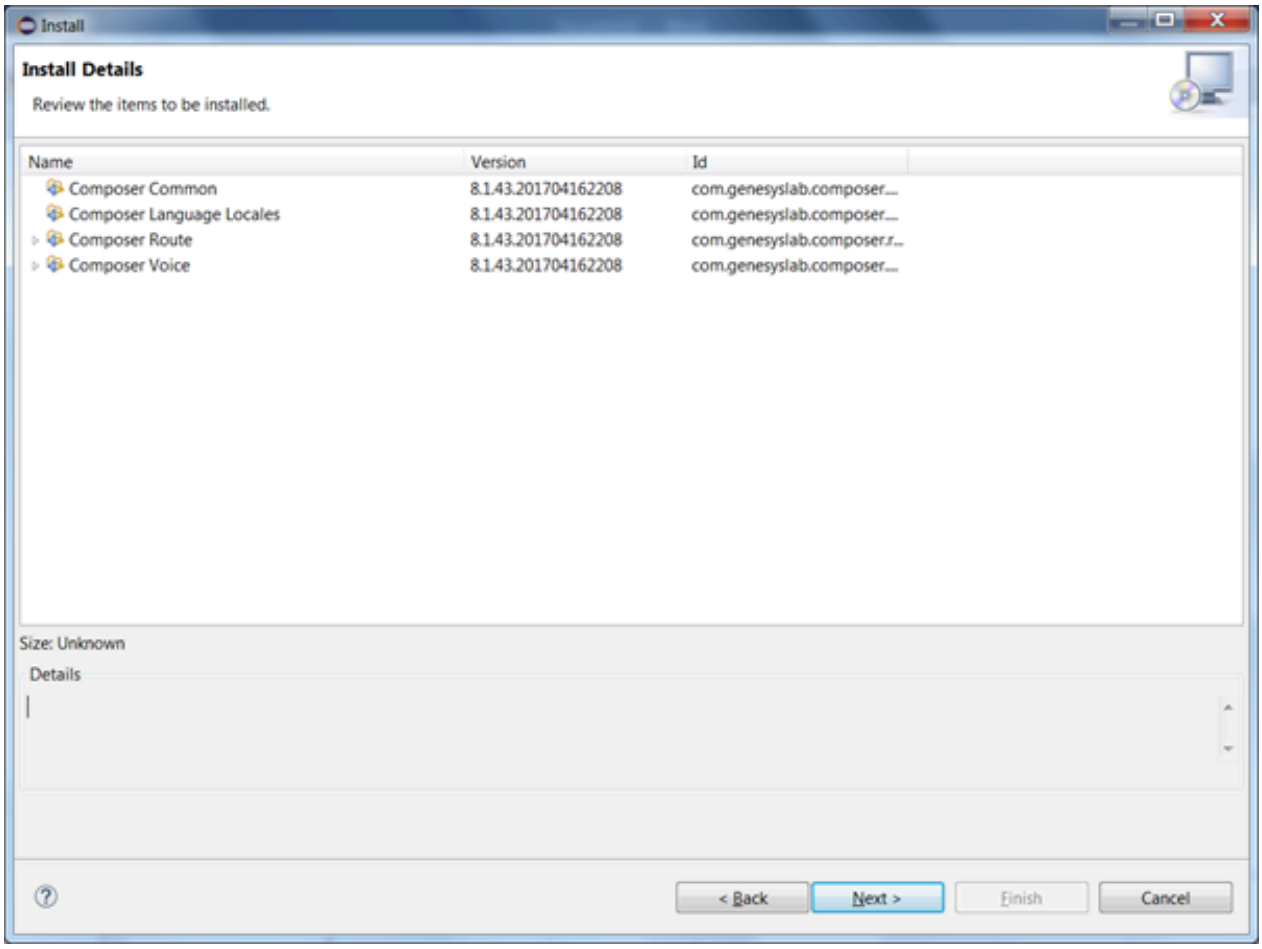
> **Important**
>
> This support is not included in previous or subsequent releases unless specifically noted in the release notes.

To display the Composer interface in French Canadian:

1. Select **Help** > **Install New Software**. (The Available Software dialog box adds the new option for Composer Language Locales, which is not shown in the installation video).
2. In the Available Software dialog box, click **Add**.
3. In the Add Repository dialog box, enter Composer and click **Archive**.
4. Navigate to the folder where `com.genesyslab.composer.updateSite.zip` is located and select it.
5. Click **Open** and then **OK**.
6. Under **Name** in the Available Software dialog box, expand **Composer**, check all the four plugin options including **Composer Language Locales** and click **Next**.



7. Review that all four plugins are selected to get installed and click **Next**.



8. Accept the License Agreement and click **Finish**.
9. Restart Eclipse to get a workspace metadata update.
10. After installation, by default the Java VM default language will be used. The Eclipse UI language can be configured by one of the following options:
    - Adding the `-nl` option at the top of the eclipse.ini file (new line for the value is must).

```
-nl
fr_CA
```

    - (OR) using `--Duser.language=fr_CA` under the -vm options in the `eclipse.ini` file.

> **Tip**
>
> If you do not have permission to save an edit to `eclipse.ini` when that file is located in a "Programs" folder, Eclipse can be installed in another folder. Composer installation prompts for the Eclipse path.

Installing Composer Offline

Starting with release 8.1.410.14, you have the option of installing Composer in an offline environment.

1. Install Composer in the Eclipse integrated development environment (IDE) on a machine that has an internet connection. Refer to the *Composer 8.1.4 Deployment Guide*, Installing Composer on Windows.
2. Make sure the Composer that is installed on Eclipse is working properly. You should be able to see the Composer features and toolbars in the Eclipse IDE.
3. Close the Eclipse Workbench and zip up the Composer installed Eclipse directory.
4. Zip the Workspace folder if it is outside the Eclipse folder or in an older Workspace from a different location. In the Workspace zip file, make sure the `.projects` folder is present inside the `workspace/.metadata/.plugins/org.eclipse.core.resources` folder.
5. Copy the Eclipse zip file to the target machine, which does not have an internet connection. Also copy the Workspace zip file from Step 4, if applicable.
6. Extract the zip files in the desired location and start `eclipse.exe`.
7. Select the Workspace folder (from step 4, if applicable). You can see the Composer features available in the copied Eclipse instance.

> **Important**
>
> Make sure the Workspace folder has USER write permissions. For more details please refer to:
>
> - Setting Up Your Workspace
> - Project Permissions

Bundled Tomcat Installation

The Composer installation procedure given above does not contain instructions for installing the bundled Tomcat server used for testing an application. Follow the steps given below to set up the Tomcat server, which is bundled with Composer.

1. Start the Composer installer again, and this time proceed by clearing (unselecting) **Install Composer Plugins**. Make sure the Composer version is the same as used in step 1 of the procedure above.
2. Copy the `composer_global.properties` file from the Composer installed installation directory and replace it with the file of the same name in the newly extracted Eclipse directory.

> **Important**
>
> This is an optional step. If Composer was installed directly on Eclipse via the **Help** > **Install New Software** option using the update site zip file, the `composer_global.properties` file may not exist in the Eclipse directory.

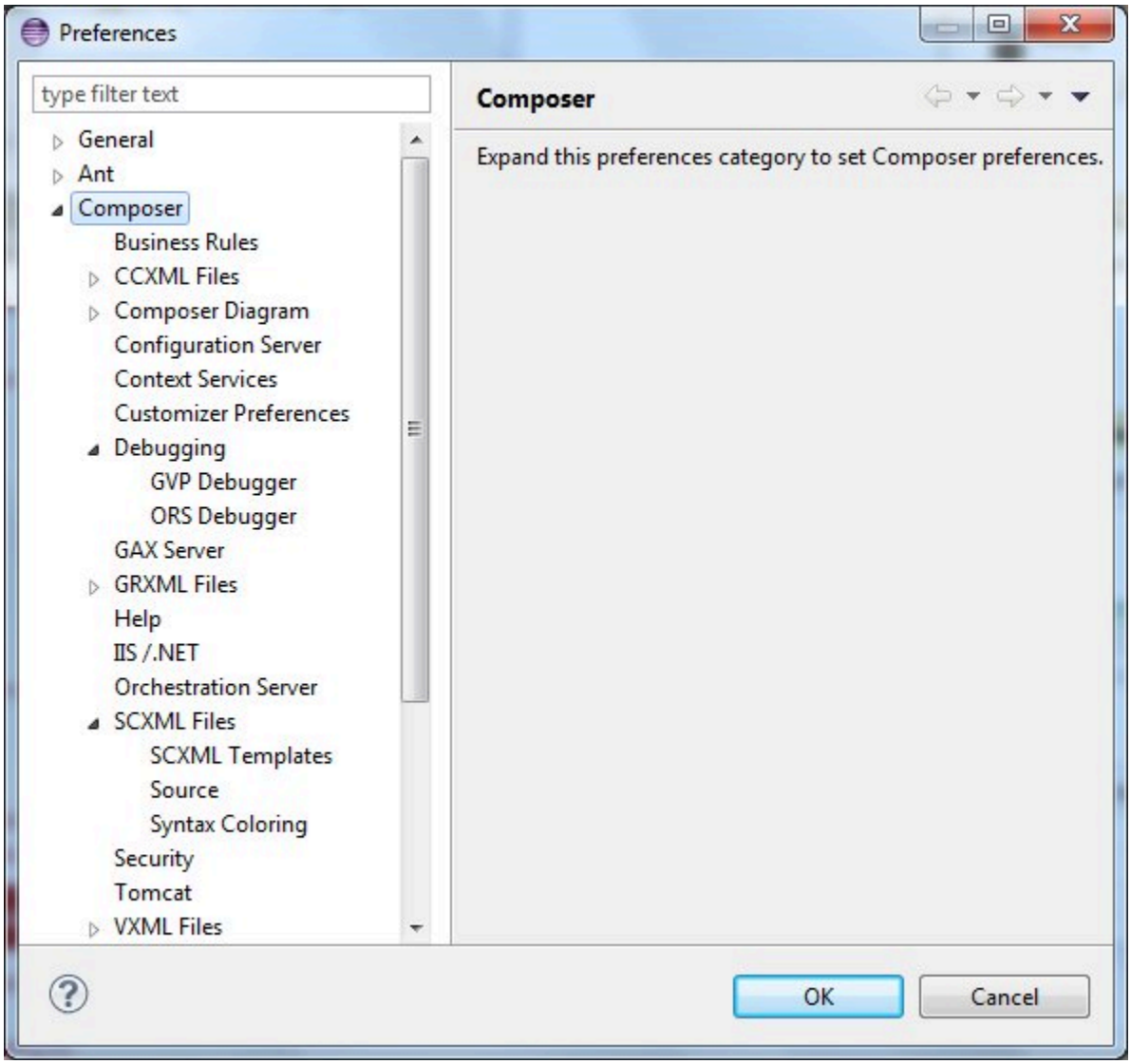
3. Start `eclipse.exe`.
4. Configure the Tomcat settings.

Post Installation Configuration

This chapter contains post-installation tasks to be performed after installing and launching Composer.

Tomcat

Before you can start to create a Java Composer Project that you will deploy later on a Tomcat application server, you must configure Tomcat settings for Composer (**Window** > **Preferences** > **Composer** > **Tomcat**). The figure below shows the Composer Preferences dialog box fully expanded.



- For information on configuring Tomcat for voice applications, see the Tomcat section in Callflow Post Installation in the Composer 8.1.4 Help.
- For information on configuring Tomcat for routing applications, see the Tomcat section in Workflow Post Installation in the Composer 8.1.4 Help.

IIS/NET Preferences

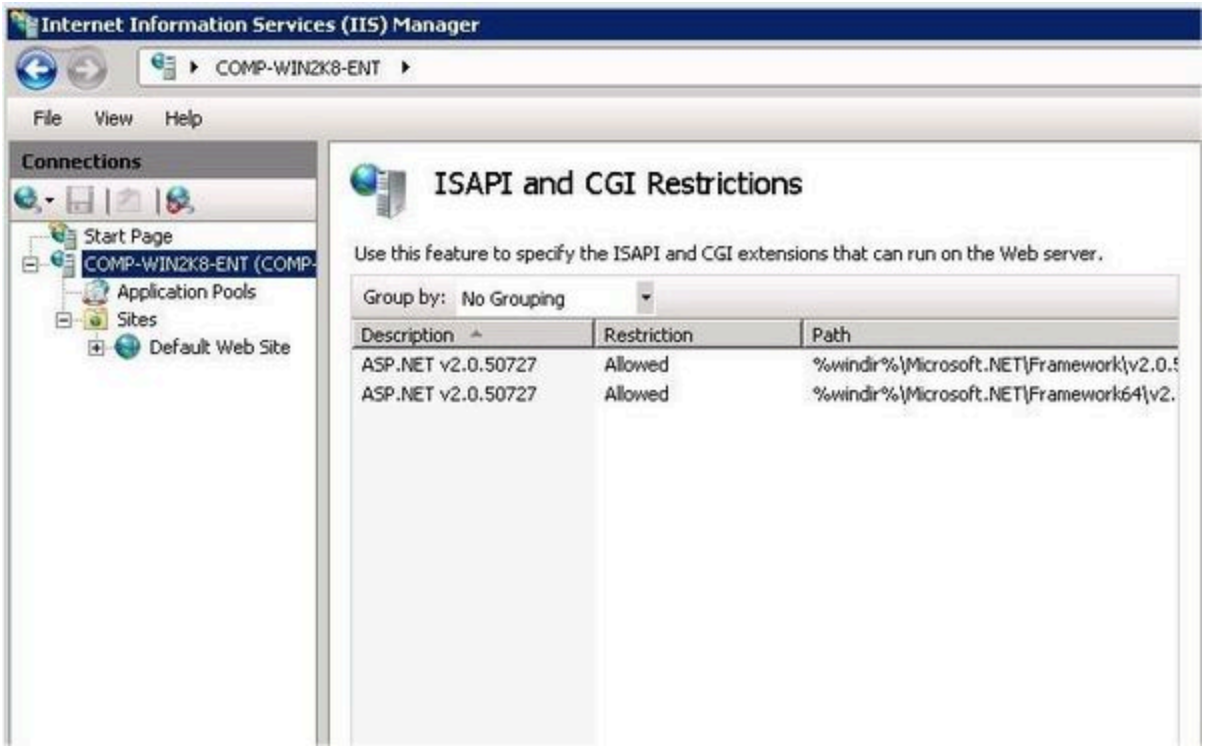
If you plan to use IIS as your web server for testing and deployment, you will also need to configure IIS preferences in Composer so that your applications can be auto-deployed to IIS from within the workbench. Composer can work only with IIS installed on the local machine. You can work with both Tomcat and IIS from the same installation of Composer.

- For more information, see Internet Information Services in Callflow Post Installation.

Internet Information Services

Before you can start to create a .NET Composer Project that you will deploy later on a Microsoft Internet Information Services (IIS) web server, you must do the following:

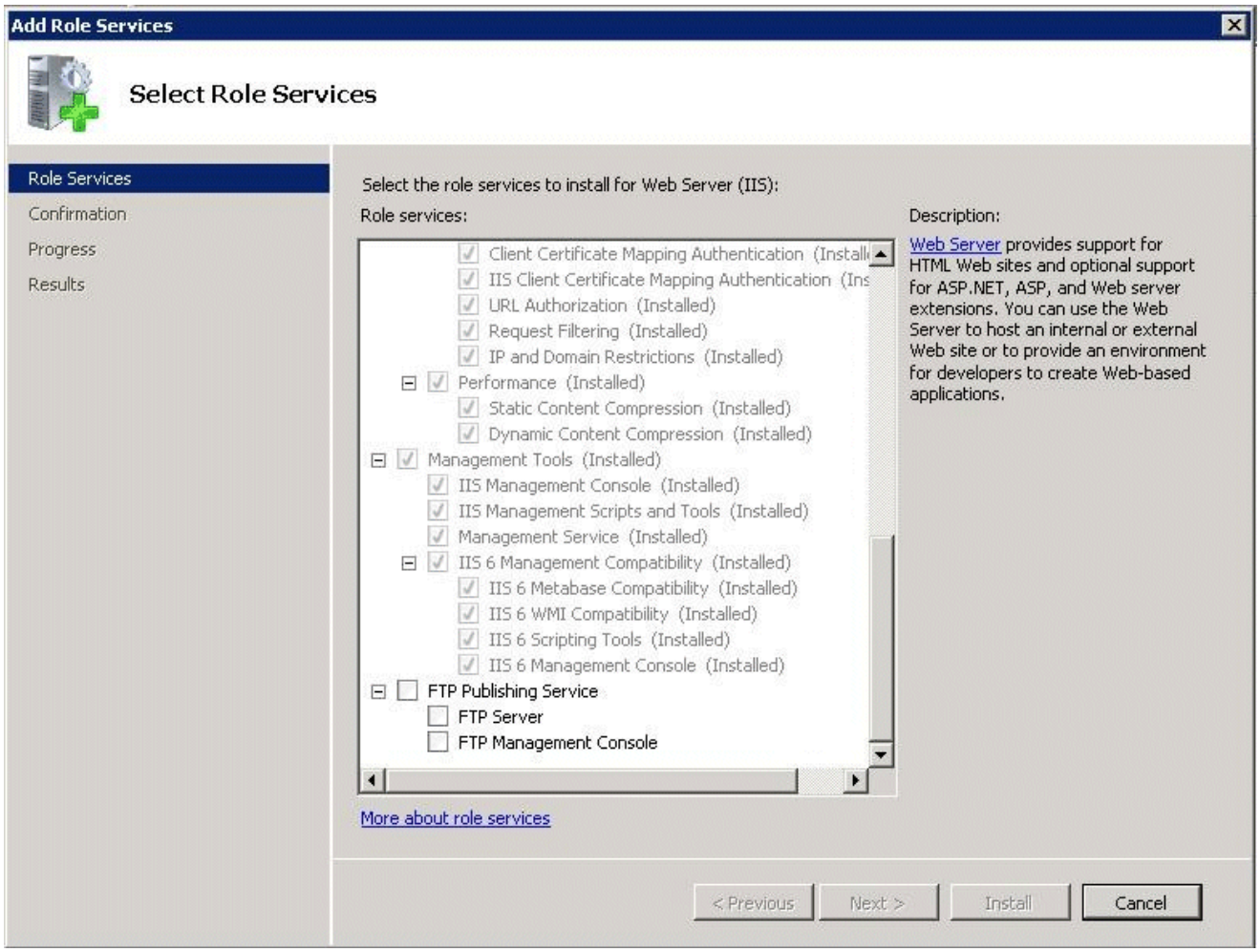
- Configure IIS/.NET Preferences (see section above).
- Allow the ASP.NET Web Service extension in Internet Information Services (IIS) Manager (see figure below).



Disabling UAC

To address any potential deployment failures when using IIS, Genesys recommends disabling the User Account Control (UAC) for all Composer supported Windows operating systems (**Control Panel** > **User Accounts** > **Use User Account Control**).

IIf running on IIS 7 (Windows Vista, Windows 2008) or IIS 7.5 (Windows 7), during the installation of IIS, the IIS Metabase and IIS 6 configuration compatibility feature must be installed. Select **Control Panel** > **Programs and Features** > **Turn Windows features on or off** > **Roles** > '**Select Role Services** (see figure below).



If this feature is not turned on, you cannot deploy a .NET Composer Project

Allowing ASP.NET Web Service Extensions in IIS

1. Open Microsoft Internet Information Services (IIS) Manager on your computer. The **Internet Information Services (IIS) Manager** window opens as shown above.
2. In the left panel, double-click the **Web Service Extensions** folder.
3. In the right panel, select the **ASP.NET** entry.
4. Exit Microsoft Internet Information Services (IIS) Manager.

WSE and .NET Projects

Microsoft Web Services Enhancements (WSE) is required for creating .NET projects in Composer. However, the WSE installer may not install on Windows 2008. These steps give a workaround:

1. Download the Microsoft WSE 3 "msi" installer bundle.
2. Use 7Zip to extract the contents to a folder.
3. In Composer, select **Window** > **Preferences** > **Composer** > **IIS/.NET** .
4. Set the **Microsoft WSE 3.0 Installed Path** field the **$Folder\Microsoft.Web.Services3.dll** file.
5. Create your Composer .NET Projects.

MIME Types

1. Open Internet Information Services (IIS) Manager on your computer.
2. Right-click your web site (such as **Default Web Site**), and select **Properties** .
3. Click the **HTTP Headers** tab.
4. Click the **MIME Types** button to display the *MIME Types* dialog box.
5. Add the following MIME types for IIS 6.0 or 7.0:

```
vox = application/octet-stream
vxml = text/xml
grxml = application/srgs+xml
wav = application/octet-stream
```

6. By default the SCXML mime type is already configured in the bundled Tomcat server. If you are using IIS you need to configure following MIME types:

```
** .json = text/json
** .scxml = text/plain
** .xml = text/xml
```

7. Make sure that ASP.NET extensions are enabled in your IIS. Right-click on the default web site and verify that the *ASP.NET* tab shows the correct version.
8. Make sure that ASP.NET is enabled on your virtual directory and set to the correct version. Right-click on the .NET Composer Project virtual directory and verify that the ASP.NET tab shows the correct version.
9. Make sure that scripts have execute permissions on your virtual directory. Right-click on the virtual directory, select **Properties** , and check the **Execute Permissions** pulldown menu. It should say **Scripts** only or **Scripts and Executables** (if you intend to run executables which is usually not done).

Debugging in MCP

To use the GVP debugging feature of Composer, the Media Control Platform (MCP) must be configured in Composer Preferences. For more information, see Callflow Post Installation, Media Control Platform section.

Configuring the GVP Debugger

In order to make test calls, you must configure (GVP) Debugger Preferences. The GVP Debugger used to debug voice applications allows you to debug an application by having the GVP Media Control Platform (MCP) initiate a call to a softphone. Once the call is answered, the MCP runs the application. You can then interact with the application just as if the call was initiated from the softphone.

For more information, see Callflow Post Installation, GVP Debugger section.

Enabling Debugging in the Media Control Platform (MCP)

To use the GVP debugging feature of Composer, the Media Control Platform (MCP) must be configured in Composer Preferences.

For more information, see Callflow Post Installation, Media Control Platform section.

Configuring the ORS Debugger

Composer provides real-time debugging capabilities for Orchestration Server (ORS) routing applications. The ORS Debugger is integrated within the workflow designer for making test calls, creating breakpoints, viewing call traces, stepping through an SCXML document/workflow, and debugging applications. In order to use the ORS Debugger, you must first set ORS Debugger Preferences.

For more information, see Workflow Post Installation, ORS Debugger sectioin.

TCP Ports (Firewalls)

If you have a local firewall on the development server (for example, Windows Firewall on Windows XP/ Windows Server 2003), make sure that the TCP ports have been opened in Composer Preferences.

For more information, see Callflow Post Installation, Firewall section.

GAX Server Preferences

GAX refers to a Genesys Administrator Extension plug-in application used by the Genesys web application, EZPulse, which enables at-a-glance views of contact center real-time statistics in the GAX user interface.

You must set GAX Preferences if you plan on using the Operational Parameter Management (OPM Block) to work with Genesys Administrator Extension. Composer diagrams connect to the GAX server using the login credentials you enter in GAX Preferences when fetching audio resource parameters.

For more information, see GAX Server Preferences.

Business Rule Preferences

Composer interfaces with the Genesys Rules Engine, which is part of the Genesys Rules System. A business rule is an external piece of business logic, which can be customized, and then invoked by Genesys applications.

You can use Composer's Business Rule block to request the Genesys Rules Engine to execute a Rule Package in a routing workflow or voice callflow and write the results back to a variable.

If you plan to use the Business Rule block, you must set Business Rule Preferences. See Business Rule Common Block, Business Rule Preferences section.

Proxy Settings

This section describes how to configure proxy settings in Tomcat and a local proxy.

Configuring Proxy Settings for a Local Proxy

If you have a local proxy on your network, you will have to configure the proxy settings to get the parsing of the Web Services Description Language (WSDL) in the Web Service block.

1. Inside Composer, configure proxy settings by going to: **Window** > **Preferences** > **General** > **Network Connections**. If necessary, provide the proxy authentication details:
2. Select **Manual proxy configuration** and provide the **HTTP proxy** and **Port** .
3. Provide the *User Name* and *Password* for authentication.
4. Click **Apply** and **OK** .

Configuring Proxy Settings in Tomcat

Starting with 8.1.420.14, Composer supports Tomcat 7. Composer installation adds the role for manager-gui to Tomcat configuration for callflows and workflows.

If you will be Deploying a Java Composer Project, Composer bundles Tomcat for running test applications, such as routing applications. Proxy settings have to be configured in the bundled Tomcat web server for the back-end pages to access the Web if you plan to use the Web Request and Web Service blocks.

1. To configure proxy settings in the bundled Tomcat web server, add the following lines to the *catalina.properties* file that is found within the *..\tomcat\conf* folder in the Composer installation path:

   http.proxyHost=hostip

   http.proxyPort=port of Proxy

http.proxyUser=username

http.proxyPassword=password

The username and password must be the same as what was provided in the manual proxy configuration

2. Restart the Tomcat service from Windows Services: **ComposerTomcat** .

Proxy Configurations for .NET Composer Projects

Each .NET Composer Project will have its own *web.config* file that needs to be updated for configuring the proxy settings.

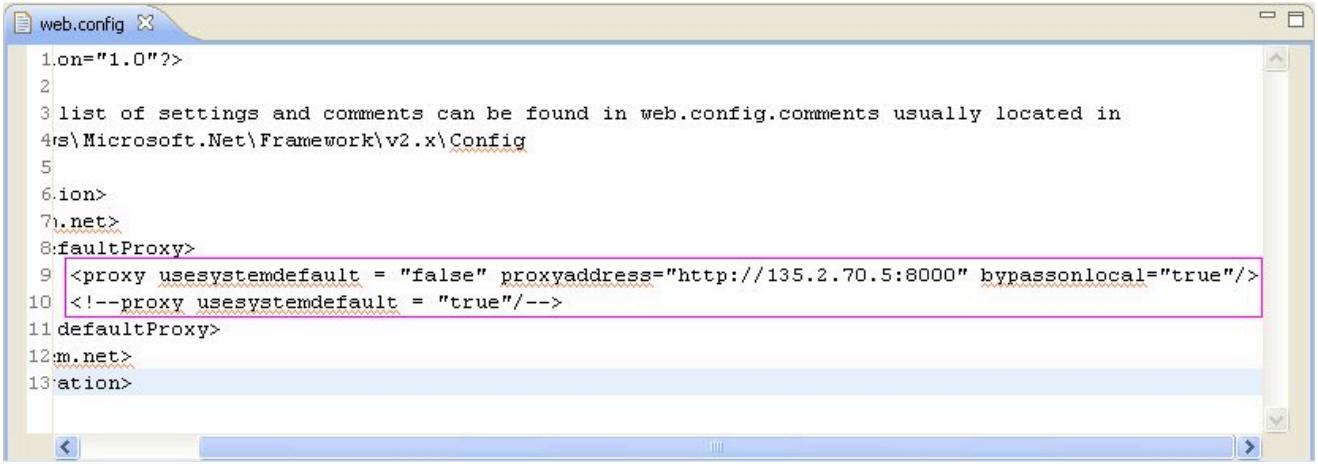
If your IIS web server is behind a proxy server and Web Request or Web Service blocks are used in a callflow, you must configure proxy settings in your .NET Composer Project. To configure proxy settings in a .NET Composer Project:

1. Open the *web.config* file.
2. Go to the <system.net> section.
3. To use the default System proxy settings: *<proxy usesystemdefault="true"/>*

If the default settings do not automatically detect the proxy server settings:

1. Set *proxy usessystemdefault* to *false* .
2. Explicitly designate the proxy server: ''<proxy usesystemdefault="false" proxyaddress="http://address" bypassonlocal="true"/>''

An example is shown below.



Prompt Resource Validation

This Composer Preference enables diagram validation warnings where prompt audio resources no longer exist in the given file path. If the audio file is no longer present, the diagram block will show a warning icon. For more information, see:

- Enable Validation for Prompt Resources in Diagram Preferences.
- Prompt Resource Validation in Callflow Post Installation.

Configuration Server Connection

When creating routing applications in Composer, you use the Configuration Database and Configuration Server. For example, when using the Target block Targets Property to route to agents or agent groups, you select targets (agents(Person), Agent Groups, and other objects) defined in the Configuration Database.

You may develop routing applications:

- With a connection to Configuration Server or
- In an "offline" mode, without connecting to Configuration Server.

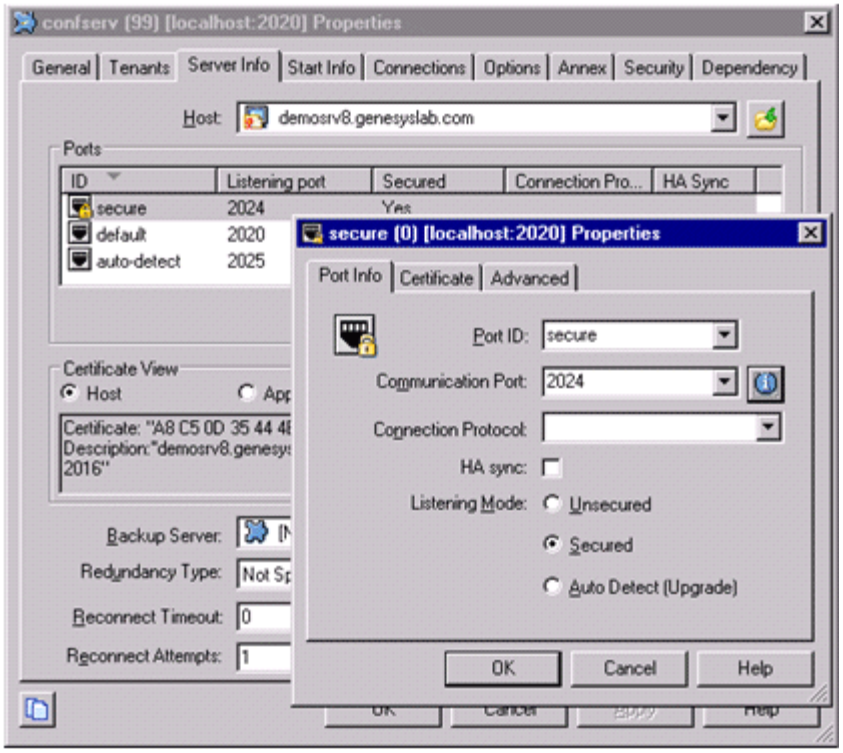
The dialog box for connecting to Configuration Server appears when you select **Connect** from the **Configuration Server** menu within Composer.

For more information, see Configuration Server in Workflow Post Installation.

TLS Support for Configuration Server

You. have the option of using a secure Transport Layer Security (TLS) connection when connecting to Configuration Server. The procedure for doing so is summarized below.

1. Generate and install certificates. See section “Certificate Generation and Installation” of the Genesys Security Deployment Guide for instructions.
2. Add a secure listening port in the Configuration Server configuration. See section “Genesys TLS Configuration” of the Genesys Security Deployment Guide for instructions. An example configuration is shown below.



3. Enable the secure connection mode when initiating the connection in Composer. In the *Connect to Configuration Server* dialog box, select *Use secure connection.*

Configuration Server Preferences

Set Configuration Server Preferences to control routing predefined statistics creation, ExpressionBuilder object validation, and Configuration Database object preferences.

Inactivity Timeout

If you have authenticated with Configuration Server, Composer times out after a configurable number of minutes of inactivity. In this case, you must reauthenticate in order to continue working with Configuration Server database objects. For Composer, inactivity is defined as a period of time with no mouse usage (click, move, and so on) or keyboard entry. For instructions on implementing this time, see the Inactivity Timeout chapter of the Genesys Security Deployment Guide.

ORS and Routing Point Configuration

When creating routing applications, in addition to specifying the HTTP request parameters, both Universal Routing Server (URS) and Orchestration Server (OS) must be properly configured.

For more information, see the Orchestration section in Workflow Post Installation.

For information on additional options that must be set, consult the following:

- Universal Routing 8.1 Deployment Guide, Orchestration Support chapter.
- Orchestration Server 8.1.4 Deployment Guide, SCXML Strategy Support and Configuring Orchestration Server chapters.

Specifying the URL of the Starting SCXML Page

1. In Genesys Administrator, in the **Provisioning** tab, select **Switching** > **Switches** .
2. Click the **DNs** tab.
3. Select the DN that corresponds to the Routing Point where the built-in root strategy is loaded.
4. In the **Annex** tab for this Routing Point, add an *orchestration* section.
5. In that section create an option named *application* and for its value enter *script:<object name>* where *<object name>* is the name of *Script* object of type *Enhanced Routing* that represents the SCXML application and contains the URL of the starting SCXML page of this application deployed on an application server. This URL needs to be accessible from the Orchestration Server host.

For complete details on this step and other routing configuration details, consult:

- The Orchestration Server 8.1.4 Deployment Guide, SCXML Application Development chapter.

Important! If you have both Composer and Interaction Routing Designer set up in the same environment, check in IRD's **Loading View** that you have not loaded an IRD routing strategy on the same Routing Point DN where the built-in strategy is loaded. This will create a conflict and cause your SCXML-based strategy not to launch.

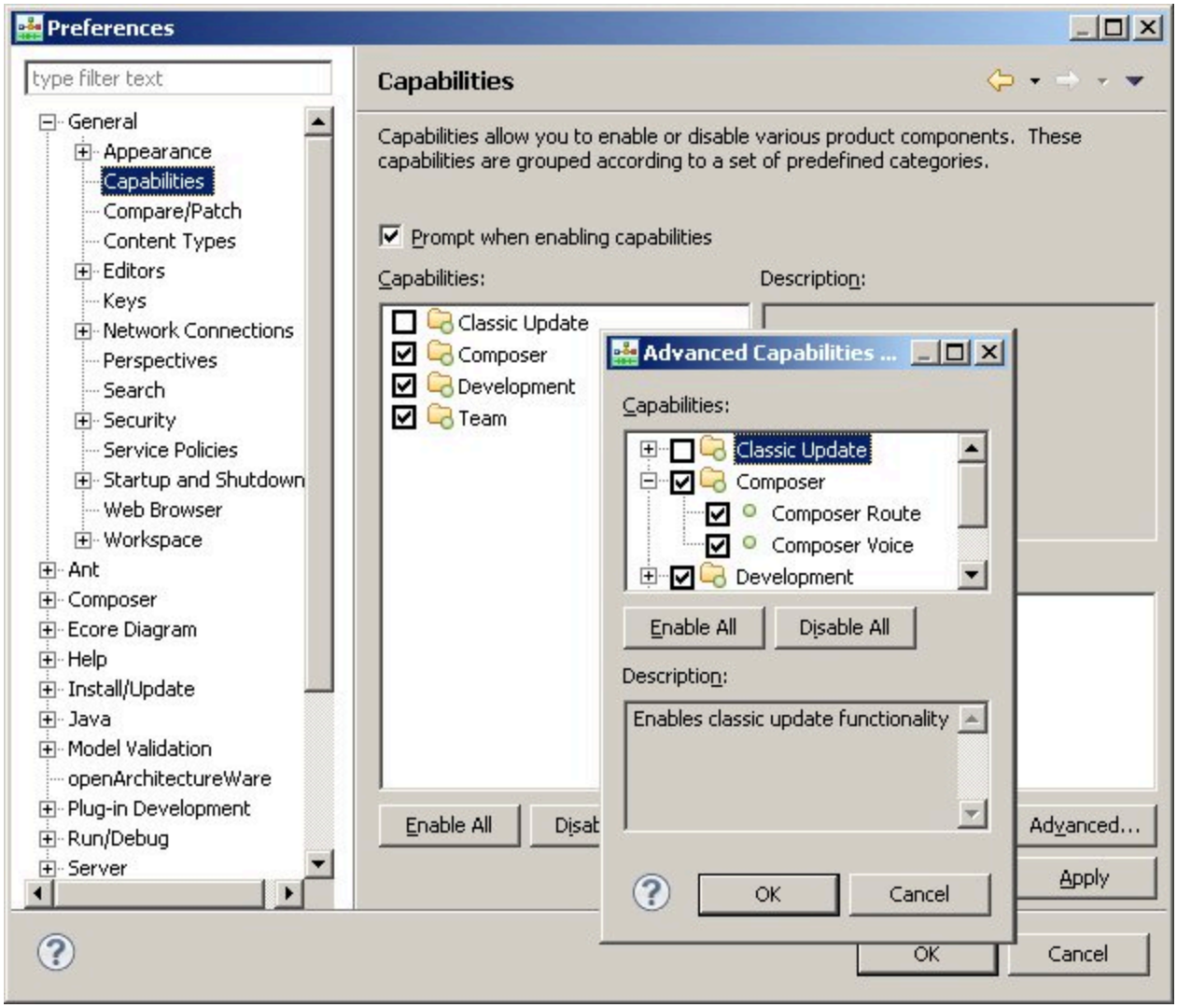
Using Stream Manager for Voice Treatments

When creating routing applications, if you plan to use Stream_Manager to play voice treatments via the Treatment blocks for workflows (such as Play Sound), additional configuration tasks are required.

For more information, see Workflow Post Installation, Stream Manager section.

Hiding Capabilities

You may hide voice or routing capabilities through a General Preference setting (**Window** > **Preference** s > **General** > **Capabilities**).



Capabilities Preferences

For more information, see Enabling-Disabling Functionality and Hiding File Types.

Context Services Preparation

Context Services refers to an optional capability of Universal Contact Server (UCS) and its UCS Database, a repository of customer-related, service, and interaction-centric data (current and historical) from Genesys and third party sources. You can use the Context Services Blocks for service personalization, offer personalization, service resumption. and enhanced reporting.

For information, see Setting Context Services Preferences.

TLS Support When Connecting to Context Services

The optional procedure below summarize how to configure a secured connection between Composer and Universal Contact Server (UCS) during application design. During runtime, the connection to UCS is initiated by Orchestration Server (ORS) or GVP.

For routing applications, no specific additional client-side configuration is needed for ORS.

For voice applications, GVP does not initiate a TLS connection to the UCS.

To use a TLS connection when using Composer to connect to UCS:

1. Generate and install a certificate in Universal Contact Server as described in Using TLS with UCS.
2. Export the certificate generated in step 1 with a command like:
   [JRE Home\bin\]keytool -export -v -alias FRBRED0H001435.emea.lucent.com -file certificate.cer-keystore certificate.jks -storepass theKeystorePassword
   Details for using the genkey command are available at the Oracle download site. See .../tooldocs/windows/keytool.html#genkeyCmd.
3. Copy the .cer certificate file generated in step 2 to the Composer host.
4. In the Composer/Security preference page, import the certificate exported in step 2.
5. In the *Composer /Context Services* preference page, enable the TLS by selecting *TLS* in the *Security Settings* area.

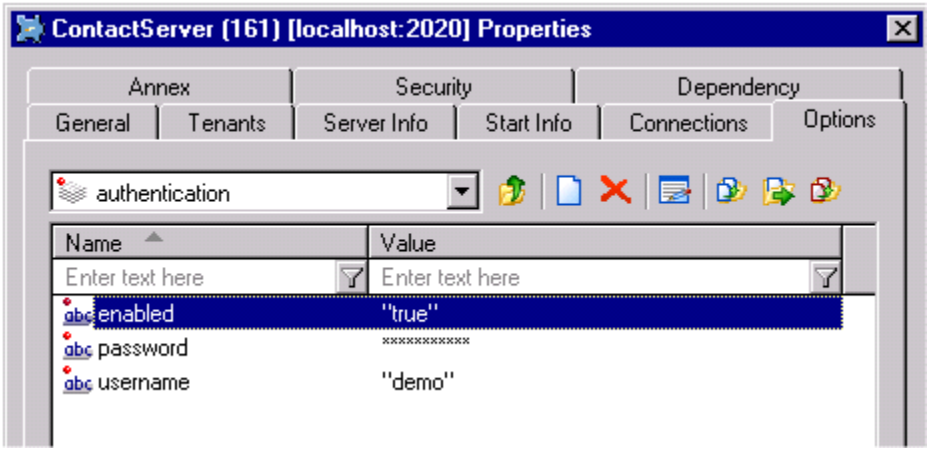
Using Context Services Authentication During Design

This section summarizes how to configure Composer to connect to Universal Contact Server using authentication when **designing** routing applications.

At this time, there is no support for authentication during **runtime**. Orchestration Server and GVP do not initiate an authenticated connection to the UCS.

To use an authenticated connection when connecting to the UCS, you need to:

1. Set the authentication mode to true in the UCS configuration.



Universal Contact Server Application, Authentication Section

See the Context Services wiki for details.

1. Open the Context Services Preferences page.
2. In the **Security Settings** area, select **Use authentication** .

Enter the username and password and click ''*Test Connection* .

Upgrading Projects and Diagrams

> **Important**
> Composer does not support upgrading diagram files from 8.0.4 versions to 8.1.2 or higher versions. If a diagram upgrade is required. first upgrade the Projects to 8.1.1 versions and then upgrade to 8.1.2 or higher versions.

While working with Composer, if you want to use a previously-created Composer Project and Project diagrams, an upgrade is required.

For more information, see Getting Started With Composer, Upgrading Projects and Diagrams.

- Genesys recommends that you create a dedicated workspace for 8.1 Projects and do not reuse previously created workspaces. This will provide a clean separation between the two versions as well as ensure that a backup copy is preserved for later reference or rollback.
- Some previously created workflow diagrams cannot be upgraded. For more information, see the Migrating IRD Strategies section below.
- Callflow diagrams designed prior to version 8.1.1 and upgraded to 8.1.1, 8.1.2, 8.1.3 and 8.1.4 versions will not have the `BlockName` variables automatically filled with the block output for the User Input blocks (Menu, Input, Record and Transfer). This implicit variable has been replaced by the `Output Result` property, where you must add/assign a variable to collect the output result of the Block.
- Starting with 8.1.410.14, the Upgrade Diagram files and Generate Code' dialog contains a **Validate** check box, which is optional. Select it if you want validation to be done after an upgrade. When this option is selected during a Project upgrade, validation will be done after generating code and the error marker will be shown in the Problems view, Project Explorer view and diagram. This validation is not applicable for a command line upgrade. Validation can be a time-consuming, especially for asp/jsp files. This option is introduced to provide backward compatibility. If you want a quick upgrade, ignore the option.
- For additional upgrade information, see the Composer 8.1.x release notes.

Migrating IRD Strategies

Composer 8.1.2 allows you to migrate some routing strategies created with Interaction Routing

Designer 8.0+ into Composer Projects as SCXML-based workflow diagrams.

For information on Composer Migration in general, see the IRD to Composer Migration Guide.

Also see the Composer chapter in the Genesys Migration Guide.

Routing Strategy Upgrade Limitations

From a routing standpoint:

- Composer 8.0.2 began support for the creation and testing of SCXML-based workflows for inbound voice use cases. Upgrading workflow diagrams created in the 8.0.2 release of Composer is therefore not supported.
- Composer 8.0.3 began support for Context Services and the processing of multimedia interactions. This release also introduced interaction process diagrams. Upgrading workflow diagrams created in the 8.0.3 release of Composer is therefore not supported.

Deploying Projects to Tomcat

If you already have a Project in the workspace and did not perform the Tomcat configuration described earlier in the Tomcat section of Workflow Post Installation, you must deploy the Project on Tomcat.

For more information, see the Testing Your Application section in Deploying Composer Applications.

Installing the Business Rules Plugin

As described in Business Rule Common Block, a Composer-compatible plug-in is available for developing Business Rule Templates.This plug-in is provided as part of the Genesys Rules System.

- To install the plugin, refer to the Genesys Rules System Deployment Guide. See Chapter 2, Installation.
- If you install the plugin, you will also have access to the Genesys Rules System Development Tool .
- Also see the Genesys Rules System Authoring Tool 8.1 Help .

Preferences and Options

Also see:

- Preferences for Voice Applications
- Preferences for Routing Applications

Hiding Data in Logs

The default log level in the Composer backend (`log4j.xml`) is set to `ERROR` so that `INFO` and `DEBUG` messages will not get printed by default.

- `INFO` level, if set, will only print the flow level statements.
- `DEBUG` level, if set, will print all the messages and also the password-related data in the log for debugging.

This applies to the the DB Data, Web Service, and Web Request blocks (Java and .NET Projects).

Uninstall and Reinstall

Exporting Projects

Important! The uninstallation may delete existing Composer projects if your Project Workspace resides in the installation directory. If this is the case, copy any existing Project folders to a safe location by following the procedure below.

Before Uninstalling, export Projects to a safe location.

1. In the Composer Project Explorer view, right click the Project folder and select **Export...**
2. In the Export dialog box, expand **General** > **File System**.
3. Click the **Next** button.
4. Check boxes for the Projects to export and choose/browse to the export destination folder.
5. Click the **Finish** button.

Uninstall Procedure

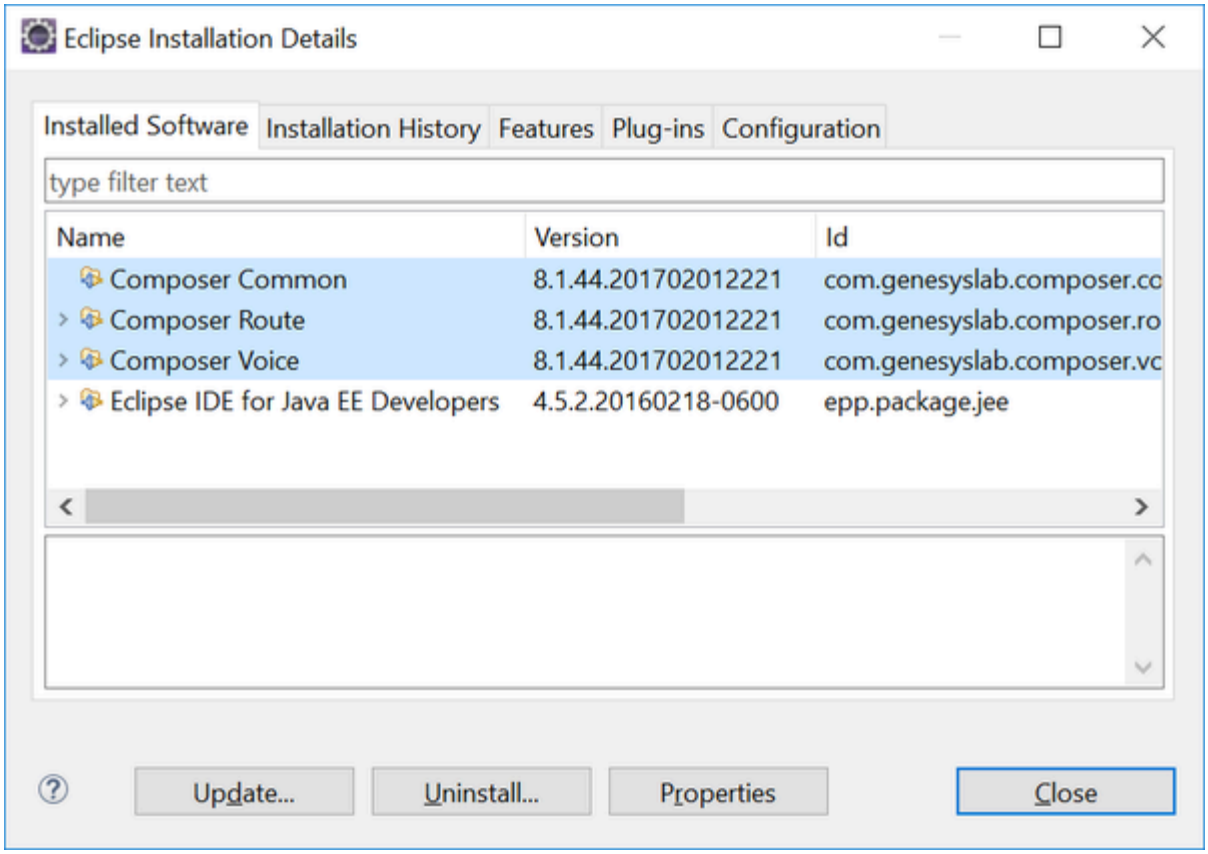
Use the procedure below to uninstall the Composer plugins in preparation for installing a newer version. A short video is presented below.

Link to video

> ### Tip
> This procedure does not involve uninstalling Eclipse or JDK. It only involves uninstalling the Composer Plugin.

To uninstall both the Eclipse plugin for Composer and the Composer installation files:

1. Right-click the Eclipse desktop icon, and select **Run as administrator**.
2. From the Eclipse Help menu, select **About Eclipse SDK**.
3. Click the **Installation Details** button.
4. Under **Installed Software**, select the Composer entries: **Composer Common**, **Composer Route**, **Composer Voice**.



5. Click **Uninstall**.
6. In your Windows Control Panel, uninstall Composer.

Reinstall Procedure

To reinstall on Windows, repeat the installation steps in Installing Composer on Windows. A short video is presented below.

Link to video

Upgrading Eclipse

When upgrading to a later version of Composer, you normally do not have to uninstall/reinstall Eclipse. You would only need to do this if you are running a version of Eclipse that is no longer supported by your Composer version. Should you wish to upgrade to a later version of Eclipse, you cannot uninstall Eclipse from the Windows Control Panel. To remove Eclipse completely, just permanently delete the unzipped folder, which will include the underlying files. Reinstalling Eclipse involves restarting, downloading the zip for the later version, extracting the contents of the zip file to the desired folder, and running `eclipse.exe` in that `eclipse` folder.

Composer Product Videos

This page contains Composer product videos. Stay tuned for more videos to come. To request a video, please email Techpubs.webadmin@genesyslab.com.

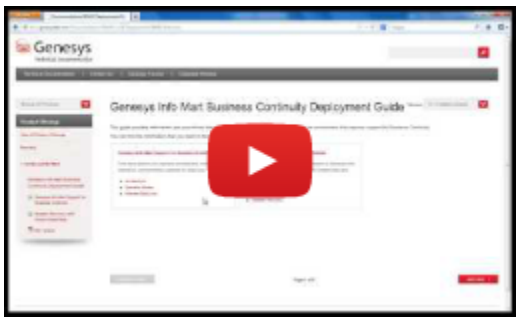
Composer Installation Video

Below is a video tutorial on Composer 8.1.4 Installation. Depending on the flavor of Eclipse you have installed, your interface may appear slightly different than that shown in the video.

Link to video

Getting Started After Installation

This tutorial shows how to immediately get familiar with Composer by using a sample application.



Uninstalling Composer

This video tutorial shows how to uninstall Composer when you want to install a later version.

Link to video

Moving to Composer from IRD

A video on using Composer to create routing strategies instead of IRD and the similarities between the two.

Link to video

Introduction to the Interface

Below is a video that can function as a brief introduction to the Composer user interface.

Link to video

Using Templates to Create a Routing Workflow

Below is a video tutorial on using Composer templates to create a workflow that routes interactions to targets based on a percent allocation.

Link to video

Integrated Voice and Route Application

This video shows an example workflow that integrates GVP voice self-service with Orchestration routing.

Link to video

Defining Agents, Agent Groups, and Skills

This video shows how Agent, Agent Group, and Skill objects are defined in Genesys Administrator prior to using them for skills-based routing in Composer.

Link to video

Skills-Based Routing

This video presents a simple example of the Composer aspect of routing chat interactions to Agent Groups. This example is based on a multimedia interaction (chat), which uses Composer's Route Interaction block, Targets property. A voice interaction uses the Target block and properties, such as the Targets property. Once ORS/URS identify a routing target, other servers are involved in the process of delivering the interaction to the agent desktop.

Link to video

Debugging VoiceXML Applications

Below is a video tutorial on debugging VoiceXML applications.

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

Link to video

Deploying a Composer Application to a Web Server

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

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

Link to video

Using the Database Blocks

Below is a video tutorial on using the Database Blocks.

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

Link to video

Creating a Simple Grammar

Below is a video tutorial on building a simple grammar with the Grammar Menu block.

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

Link to video

Using the Web Service Block

Below is a video tutorial on using the Web Service block.

> **Tip**
> While the interface for Composer in this video is from release 8.0.1, the steps are the basically the same for subsequent releases.