Genesys®

Composer 8.1

Deployment Guide

About Genesys

Genesys is the world's leading provider of customer service and contact center software—with more than 4,000 customers in 80 countries. Drawing on its more than 20 years of customer service innovation and experience, Genesys is uniquely positioned to help companies bring their people, insights and customer channels together to effectively drive today's customer conversation. Genesys software directs more than 100 million interactions every day, maximizing the value of customer engagement and differentiating the experience by driving personalization and multi-channel customer service—and extending customer service across the enterprise to optimize processes and the performance of customer-facing employees. Go to www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support. Before contacting technical support, please refer to the ***Genesys Care Program Guide*** for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys Licensing Guide.***

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 812composer_dep_01-2013_v8.1.201.00

Genesys

Table of Contents

List of Procedures

Preface

Welcome to the *Composer 8.1 Deployment Guide.* This guide describes how to deploy Composer, an Integrated Development Environment (IDE) used to develop applications for:

- Genesys Voice Platform (GVP)
- Orchestration Server (ORS) Platform

This document is valid only for the 8.1.2 release of this product.

Note: The 8.0.1 version of this product was known as Composer Voice, as it was used only to develop applications for GVP. For versions of this document created for Composer Voice, visit the Genesys Technical Support website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesyslab.com`.

This preface contains the following sections:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 73.

About Composer

An Eclipse-based application (www.eclipse.org), Composer provides both drag-and-drop graphical development of voice applications (or “callflows”) and routing strategies (or “workflows”) as well as syntax-directed editing of these applications.

- For GVP applications, Composer supports editing of VoiceXML 2.1, CCXML1.0, and SRGS 1.0.
- For ORS applications, Composer supports editing of SCXML 1.0.

Composer provides real-time debugging capabilities for both GVP voice applications and ORS routing applications.

- The GVP debugger is integrated with GVP for making test calls, viewing call traces, and debugging applications.
- The ORS debugger is integrated with ORS for making test calls, creating breakpoints, viewing call traces, and stepping through an SCXML document/workflow.

For more information on Composer, see Chapter 2, “Overview” on page 13. For step-by-step instructions on using Composer, see the *Composer 8.1 Help*.

Intended Audience

This document is primarily intended for system integrator 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 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 to when building applications.

Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to Techpubs.webadmin@genesyslab.com.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the scope of this document only and to the way in which the information is presented. Contact your Genesys Account Representative or Genesys Technical Support if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Contacting Genesys Technical Support

If you have purchased support directly from Genesys, please contact Genesys Technical Support.

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

Document Change History

This section lists topics that are new or have changed significantly since the 8.0 release of this document.

New in Document Version 8.1.201.00

Updated screen captures to reflect 8.1.2 in "Overview" on page 13.

Added new 8.1.2 features in "New Features in 8.1" on page 27.

Updated "Genesys Software Prerequisites" on page 42.

Added references to the Composer 8.1 Help Wiki in Chapter 3 on page 51.

Added "Configuring GAX Server Preferences" on page 58.

New in Document Version 8.1.101.00

- Added ORS Debugger to "About Composer" on page 10.
- Updated "Available Perspectives" on page 17.
- Added Orchestration Server wiki in "Expression Builder" on page 18.
- Added "Debugging Routing SCXML Applications" on page 22.

- Updated “New Features in 8.1” on page 27.
- Added note under “Genesys Software Prerequisites” on page 42 regarding Orchestration Server component requirements.
- Added “Configuring the ORS Debugger” on page 57.
- Added additional section in *Orchestration Server Deployment Guide* to “Specifying the URL of the Starting SCXML Page” on page 64.

New in Document Version 8.1.001.00

- Updated statement about installing IIS Metabase and IIS 6 configuration compatibility and disabling Windows User Account Control. See the notes on page 53.
- Updated “New Features in 8.1” on page 27.
- Added capability of migrating IRD strategies into Composer. See “Migrating IRD Strategies” on page 68.
- Added “Installing the Business Rules Plugin” on page 69.
- Updated step 5 in the procedure “Specifying the URL of the Starting SCXML Page” on page 64.
- Under “Third-Party Software Requirements” on page 41, added 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.
- Under “Database Support” on page 40, added that Composer 8.1 has switched to an Oracle Provider from a Microsoft Provider for .NET Composer projects.
- Under “Security Banner Configuration” on page 43, added that there is no Application object in Configuration Server for Composer. The security banner configuration is read from the registry.
- Updated versions under “Genesys Software Prerequisites” on page 42.
- To avoid duplication, many of the procedures previously in Chapter 3 on page 51 now refer to the exact location in the *Composer 8.1 Help*.
- Replaced *Genesys 8.0 SCXML Technical Reference* with “Orchestration Server Wiki” on page 73.

Chapter

1 Overview

This chapter summarizes Composer's main features, shows examples of its diagram designer, code editor, and debugging interface, and describes the new features in each release. This chapter contains the following sections:

What is Composer?

Composer is an Integrated Development Environment, based on Eclipse 3.5.1. To familiarize yourself with basic Eclipse concepts, refer to the *Workbench User Guide* by selecting `Help` > `Help Contents` in the Composer main window and expanding the `Workbench User Guide` link.

You can use Composer to develop:

- Voice applications for Genesys Voice Platform (GVP) 8.x —a software suite that unifies voice and web technologies to provide a complete solution for customer self-service or assisted service.
- Routing applications for the Orchestration 8.x platform—Orchestration Server (ORS) is responsible for executing orchestration logic (SCXML) that is provided by an application server, such as an application server hosting a SCXML-based routing application created in Composer. The responsibility of the Universal Routing Server within the Orchestration Platform is to provide a necessary service to Orchestration Server, to support routing functions.

Composer provides a rich development experience, which Web Application developers are already used to, for building VoiceXML, CCXML, GRXML, and SCXML applications.

Applications That Can be Developed

Composer provides ability to develop the following types of applications.

For the Genesys Voice Platform (GVP) 8.x Next Generation Interpreter (NGI):

- VoiceXML Applications with full support for Genesys extensions.
- CCXML + VXML Applications requiring advanced call control features including conferencing.

Note: To develop a CCXML application, you invoke an editor that performs syntax checking and enter the code. With VXML or SCXML applications, you can invoke a rich editor or you have the option of working with blocks (see Figure 1).

- CTI + VXML Applications for Genesys Framework.

For the Orchestration Server 8.x SCXML Engine/Interpreter:

- SCXML Applications with full support for all Genesys predefined SCXML functional modules and extensions used for creating SCXML-based routing workflows for both voice and non-voice (multimedia) interactions.

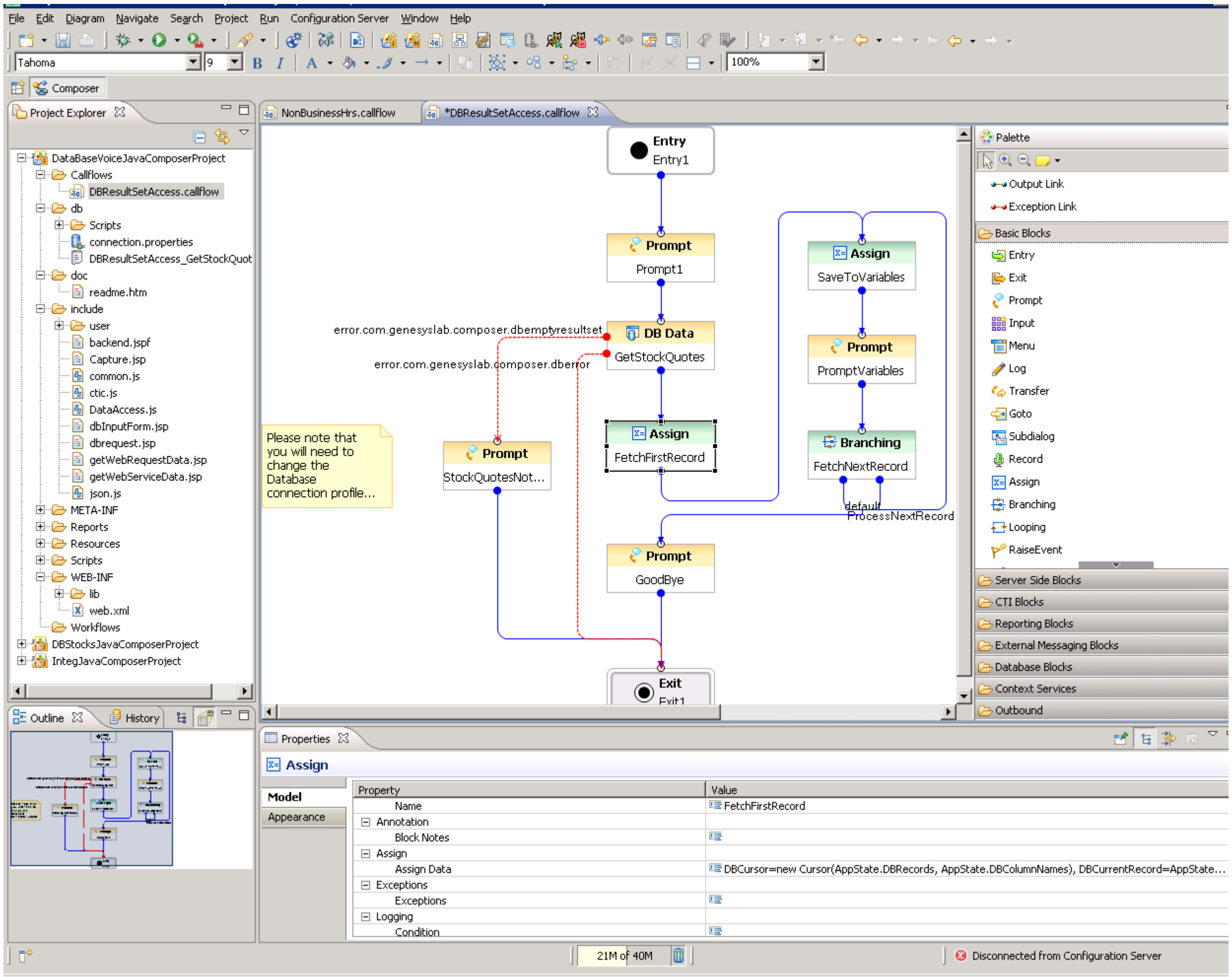
Composer also provides Integrated CTI + VoiceXML applications for end-to-end treatment handling in conjunction with GVP and Stream Manager.

Composer 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 can easily create flow diagrams by placing and connecting blocks and configuring properties. Figure 1 shows an example callflow in the center editing area.



Please note that you will need to change the Database connection profile...

Property	Value
Name	FetchFirstRecord
⊟ Annotation	
Block Notes	
⊟ Assign	
Assign Data	DBCursor=new Cursor(AppState.DBRecords, AppState.DBColumnNames), DBCurrentRecord=AppState...
⊟ Exceptions	
Exceptions	
⊟ Logging	
Condition	

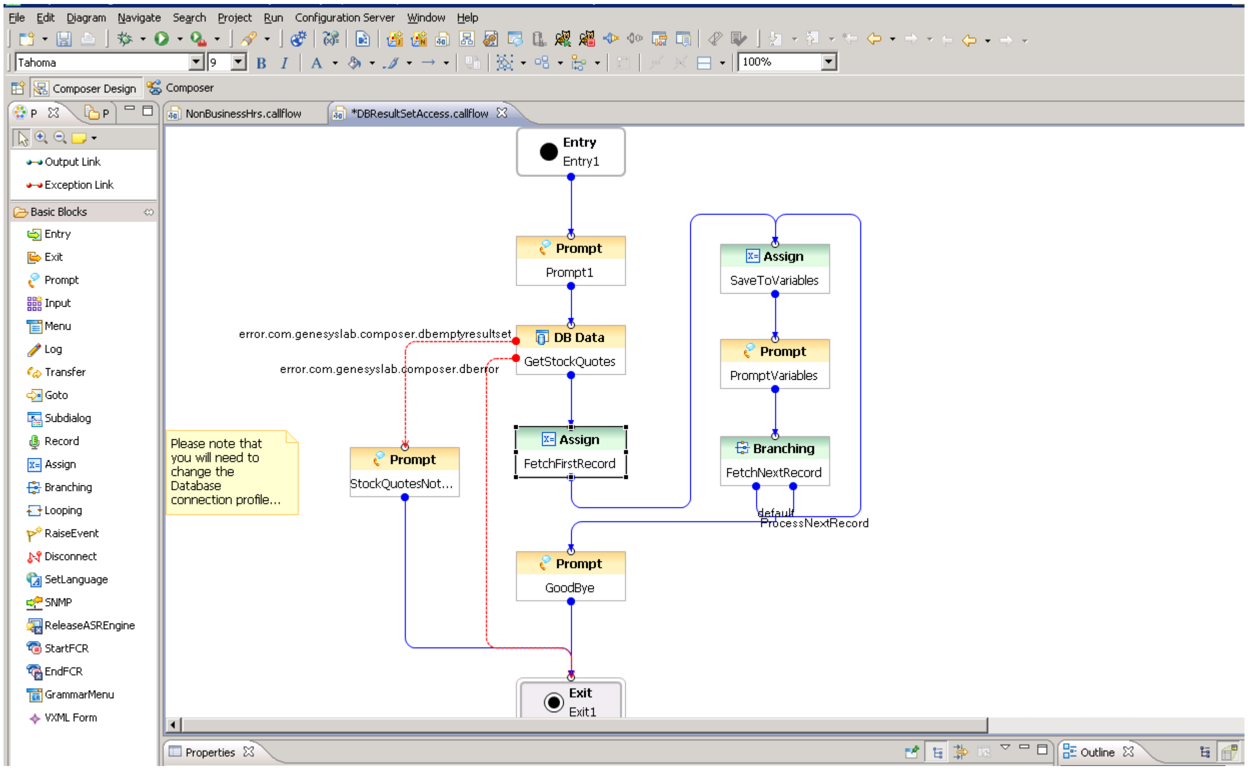
Figure 1: Callflow in Composer Perspective

Composer 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.

Figure 1 above shows Composer perspective. Figure 2 below shows Composer Design perspective, which maximizes the design area. Having a larger design area is useful when creating 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.



Please note that you will need to change the Database connection profile...

Figure 2: Callflow in Composer Design Perspective

The interface elements in Composer perspective (Figure 1 on page 15) 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, 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 view for configuring properties, along with other views upon demand.

- A “palette” of blocks on the upper right for creating flow diagrams.

Views in Lower Pane

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`

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

- `Properties`
- `Problems`
- `List Objects Manager`
- `Statistics Manager`.

Notes: The perspective for building callflows contains the same interface elements as the perspective for building workflows.

You can move different views/tabs around to customize the perspective to suit your needs, and also revert to the default perspective if desired.

Available Perspectives

Composer includes the following perspectives for building applications:

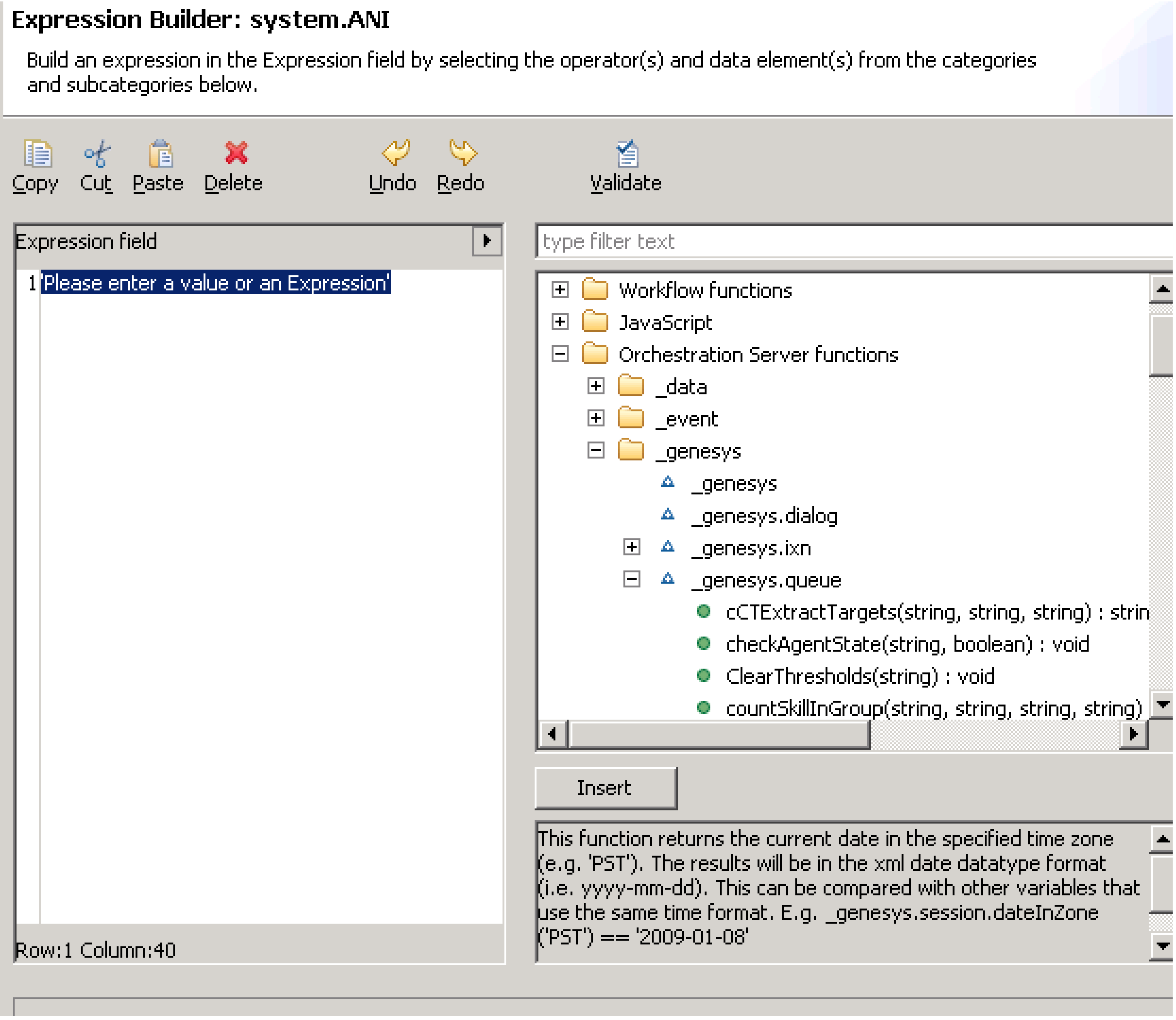
- `GVP Debugger`, for debugging voice callflows that you build or import.
- `ORS Debugger`, for debugging routing workflows that you build or import.
- `Prompts Manager`, which provides the ability to quickly review all prompts in a Composer Project.
- `Composer`, for both voice and routing applications,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

Expression Builder

Composer supplies Expression Builder to easily build expressions that can be used for branching and conditional routing decisions. You can also build ECMAScript expressions that use Genesys supplied SCXML functions documented in the *SCXML Technical Reference* (available from `Help` > `Contents`) and also in the `Orchestration Server wiki`, which is the Genesys language specification.

Figure 3 shows how to access the Genesys-supplied functions.



Expression Builder: system.ANI

Build an expression in the Expression field by selecting the operator(s) and data element(s) from the categories and subcategories below.

Expression field

'Please enter a value or an Expression'

Row:1 Column:40

type filter text

This function returns the current date in the specified time zone (e.g. 'PST'). The results will be in the xml date datatype format (i.e. yyyy-mm-dd). This can be compared with other variables that use the same time format. E.g. _genesys.session.dateInZone ('PST') == '2009-01-08'

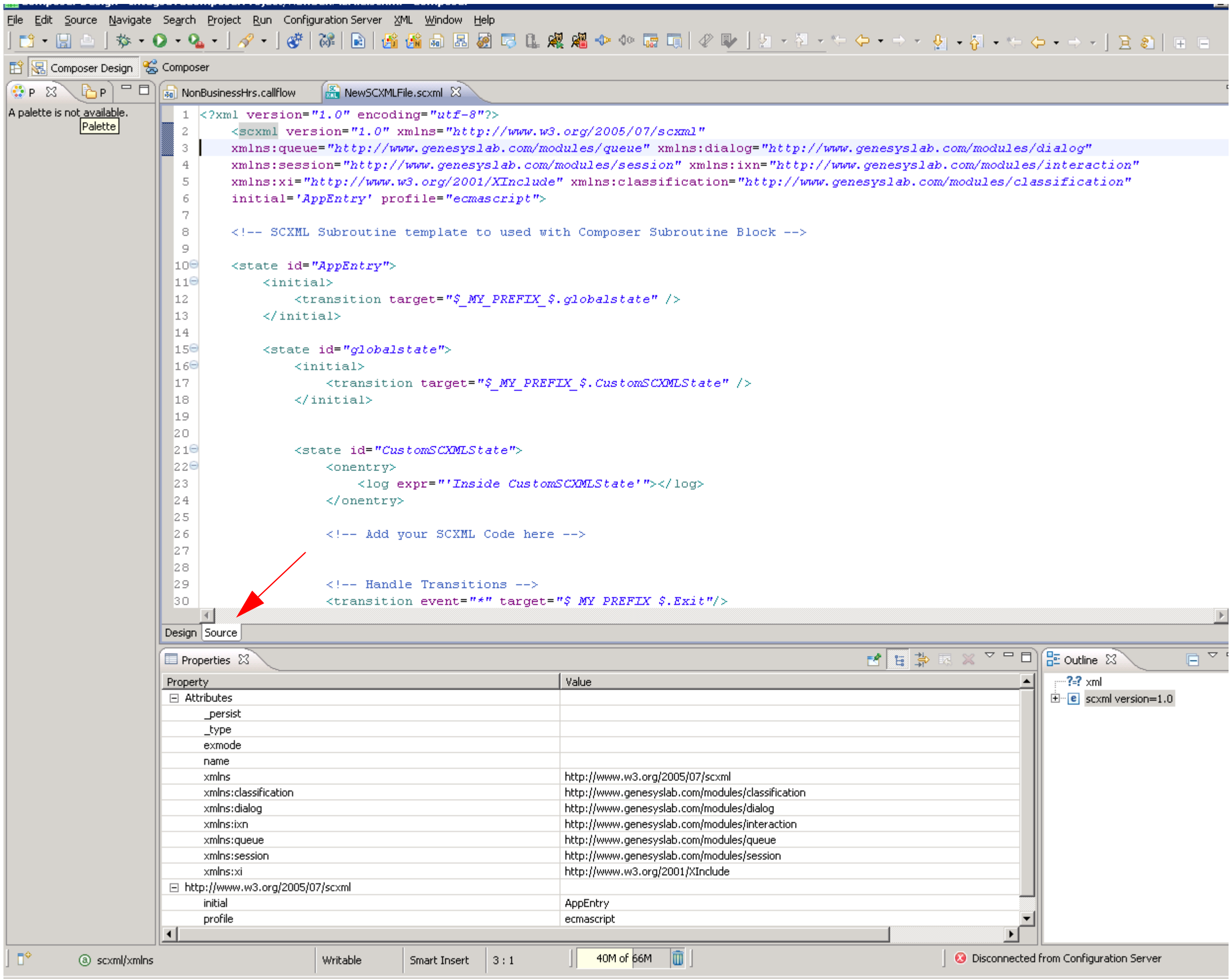
Copy
Cut
Paste
Delete
Undo
Redo
Validate
Workflow functions
JavaScript
Orchestration Server functions
_data
_event
_genesys
_genesys
_genesys.dialog
_genesys.ixn
_genesys.queue
cCTExtractTargets(string, string, string) : strin
checkAgentState(string, boolean) : void
ClearThresholds(string) : void
countSkillInGroup(string, string, string, string)
Insert

Figure 3: 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.

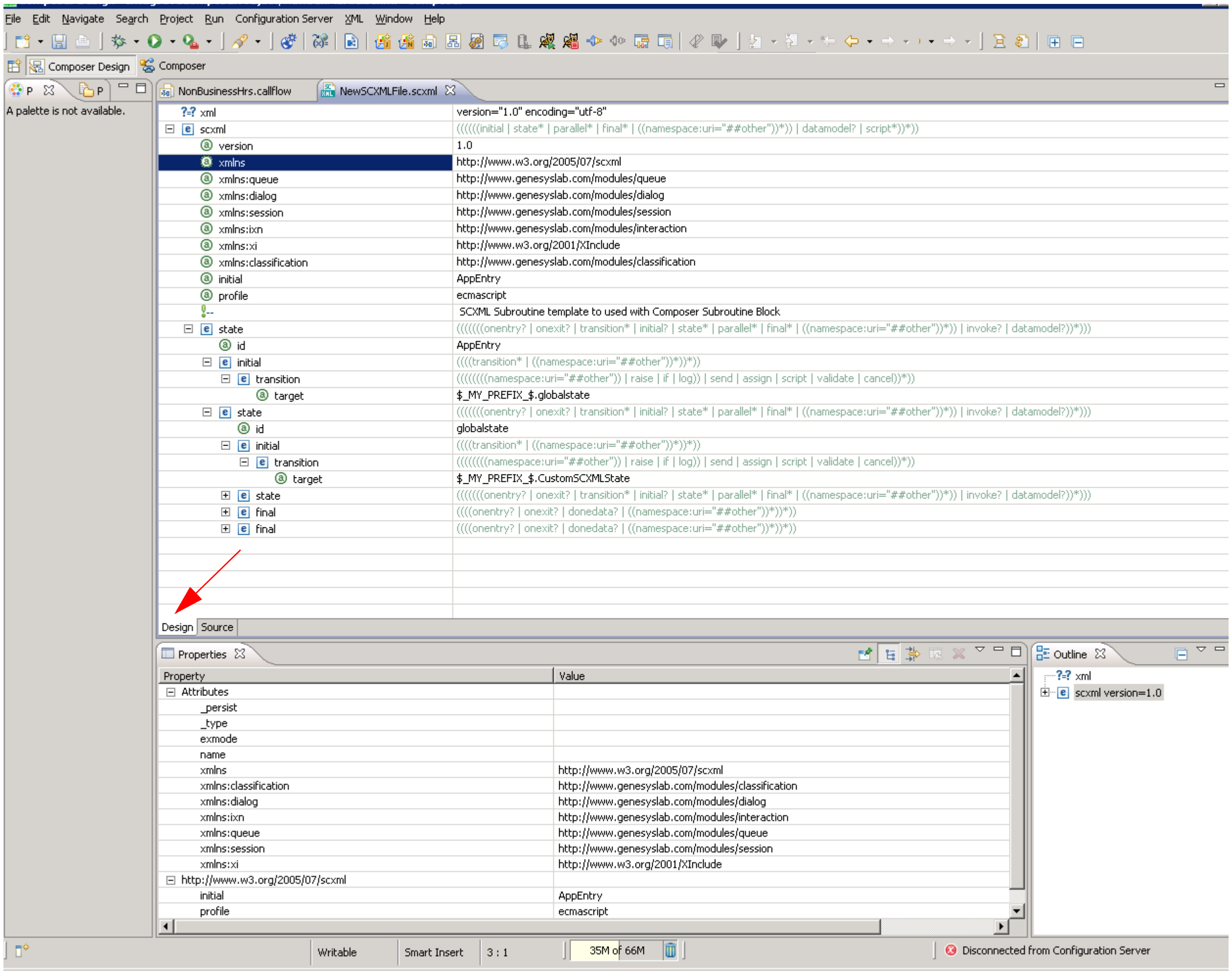
- Figure 4 shows example SCXML code in the `Source` tab of the editor.
- Figure 5 shows the `Design` tab of the editor.



```
<?xml version="1.0" encoding="utf-8"?>
    <scxml version="1.0" xmlns="http://www.w3.org/2005/07/scxml"
    xmlns:queue="http://www.genesyslab.com/modules/queue" xmlns:dialog="http://www.genesyslab.com/modules/dialog"
    xmlns:session="http://www.genesyslab.com/modules/session" xmlns:ixn="http://www.genesyslab.com/modules/interaction"
    xmlns:xi="http://www.w3.org/2001/XInclude" xmlns:classification="http://www.genesyslab.com/modules/classification"
    initial='AppEntry' profile="ecmascript">

    <!-- SCXML Subroutine template to used with Composer Subroutine Block -->

    <state id="AppEntry">
        <initial>
            <transition target="$_MY_PREFIX_$.globalstate" />
        </initial>

        <state id="globalstate">
            <initial>
                <transition target="$_MY_PREFIX_$.CustomSCXMLState" />
            </initial>


            <state id="CustomSCXMLState">
                <onentry>
                    <log expr="'Inside CustomSCXMLState'"></log>
                </onentry>

                <!-- Add your SCXML Code here -->


                <!-- Handle Transitions -->
                <transition event="*" target="$_MY_PREFIX_$.Exit"/>
```

Property	Value
⊟ Attributes	
_persist	
_type	
exmode	
name	
xmlns	http://www.w3.org/2005/07/scxml
xmlns:classification	http://www.genesyslab.com/modules/classification
xmlns:dialog	http://www.genesyslab.com/modules/dialog
xmlns:ixn	http://www.genesyslab.com/modules/interaction
xmlns:queue	http://www.genesyslab.com/modules/queue
xmlns:session	http://www.genesyslab.com/modules/session
xmlns:xi	http://www.w3.org/2001/XInclude
⊟ http://www.w3.org/2005/07/scxml	
initial	AppEntry
profile	ecmascript

Figure 4: Composer Code Editor, Source View



?-? xml	version="1.0" encoding="utf-8"
scxml	((((((initial | state* | parallel* | final* | ((namespace:uri="##other"))*)) | datamodel? | script*))*))
version	1.0
xmlns	http://www.w3.org/2005/07/scxml
xmlns:queue	http://www.genesyslab.com/modules/queue
xmlns:dialog	http://www.genesyslab.com/modules/dialog
xmlns:session	http://www.genesyslab.com/modules/session
xmlns:ixn	http://www.genesyslab.com/modules/interaction
xmlns:xi	http://www.w3.org/2001/XInclude
xmlns:classification	http://www.genesyslab.com/modules/classification
initial	AppEntry
profile	ecmascript
!--	SCXML Subroutine template to used with Composer Subroutine Block
state	(((((((onentry? | onexit? | transition* | initial? | state* | parallel* | final* | ((namespace:uri="##other"))*)) | invoke? | datamodel?))*)))
id	AppEntry
initial	((((transition* | ((namespace:uri="##other"))*))*))
transition	((((((((namespace:uri="##other")) | raise | if | log)) | send | assign | script | validate | cancel))*))
target	$_MY_PREFIX_$.globalstate
state	(((((((onentry? | onexit? | transition* | initial? | state* | parallel* | final* | ((namespace:uri="##other"))*)) | invoke? | datamodel?))*)))
id	globalstate
initial	((((transition* | ((namespace:uri="##other"))*))*))
transition	((((((((namespace:uri="##other")) | raise | if | log)) | send | assign | script | validate | cancel))*))
target	$_MY_PREFIX_$.CustomSCXMLState
state	(((((((onentry? | onexit? | transition* | initial? | state* | parallel* | final* | ((namespace:uri="##other"))*)) | invoke? | datamodel?))*)))
final	((((onentry? | onexit? | donedata? | ((namespace:uri="##other"))*))*))
final	((((onentry? | onexit? | donedata? | ((namespace:uri="##other"))*))*))

Property	Value
Attributes	
_persist	
_type	
exmode	
name	
xmlns	http://www.w3.org/2005/07/scxml
xmlns:classification	http://www.genesyslab.com/modules/classification
xmlns:dialog	http://www.genesyslab.com/modules/dialog
xmlns:ixn	http://www.genesyslab.com/modules/interaction
xmlns:queue	http://www.genesyslab.com/modules/queue
xmlns:session	http://www.genesyslab.com/modules/session
xmlns:xi	http://www.w3.org/2001/XInclude
http://www.w3.org/2005/07/scxml	
initial	AppEntry
profile	ecmascript

Figure 5: Composer Code Editor, Design View

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.

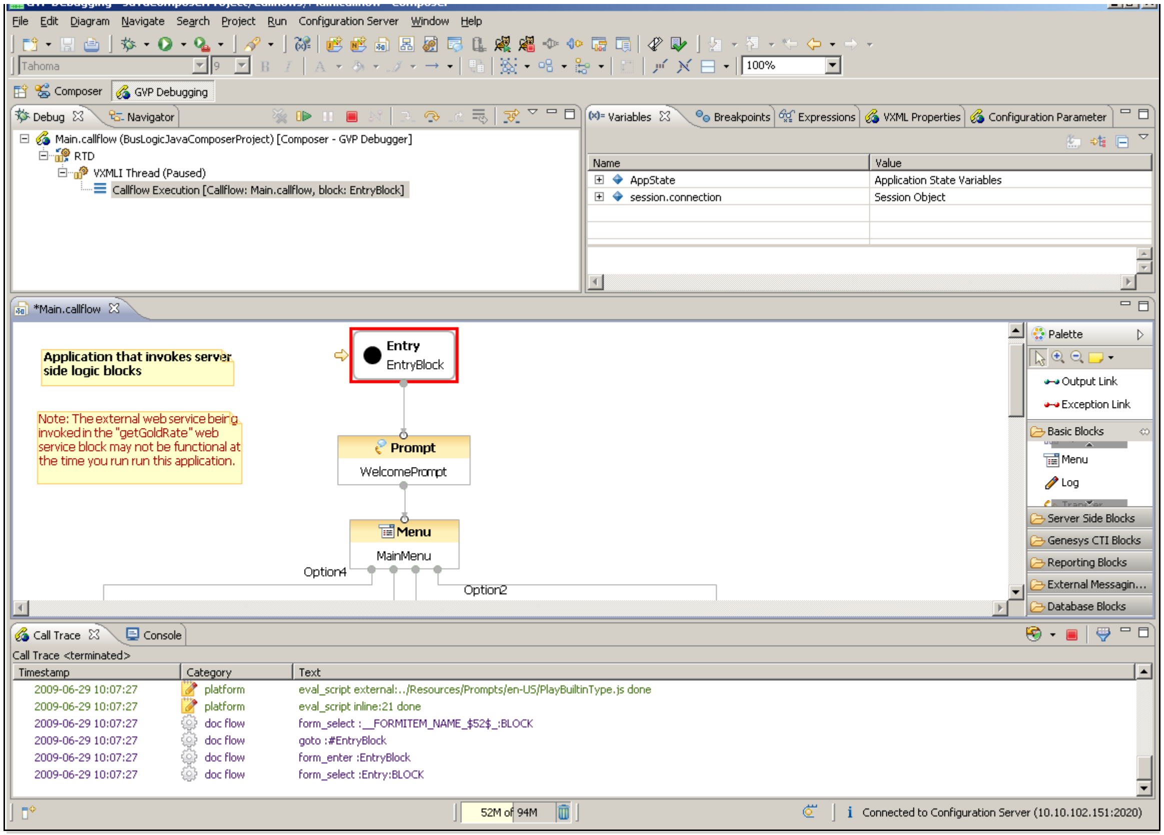
Debugging VoiceXML Applications

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. Figure 6 shows a callflow in GVP Debugging perspective.



Main.callflow (BusLogicJavaComposerProject) [Composer - GVP Debugger]
RTD
VXMLI Thread (Paused)
Callflow Execution [Callflow: Main.callflow, block: EntryBlock]

Name	Value
AppState	Application State Variables
session.connection	Session Object

Application that invokes server side logic blocks

Note: The external web service being invoked in the "getGoldRate" web service block may not be functional at the time you run run this application.

Call Trace <terminated>

Timestamp	Category	Text
2009-06-29 10:07:27	platform	eval_script external:../Resources/Prompts/en-US/PlayBuiltinType.js done
2009-06-29 10:07:27	platform	eval_script inline:21 done
2009-06-29 10:07:27	doc flow	form_select :__FORMITEM_NAME_52_:BLOCK
2009-06-29 10:07:27	doc flow	goto :#EntryBlock
2009-06-29 10:07:27	doc flow	form_enter :EntryBlock
2009-06-29 10:07:27	doc flow	form_select :Entry:BLOCK

Connected to Configuration Server (10.10.102.151:2020)

Figure 6: GVP Debugging Perspective

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.

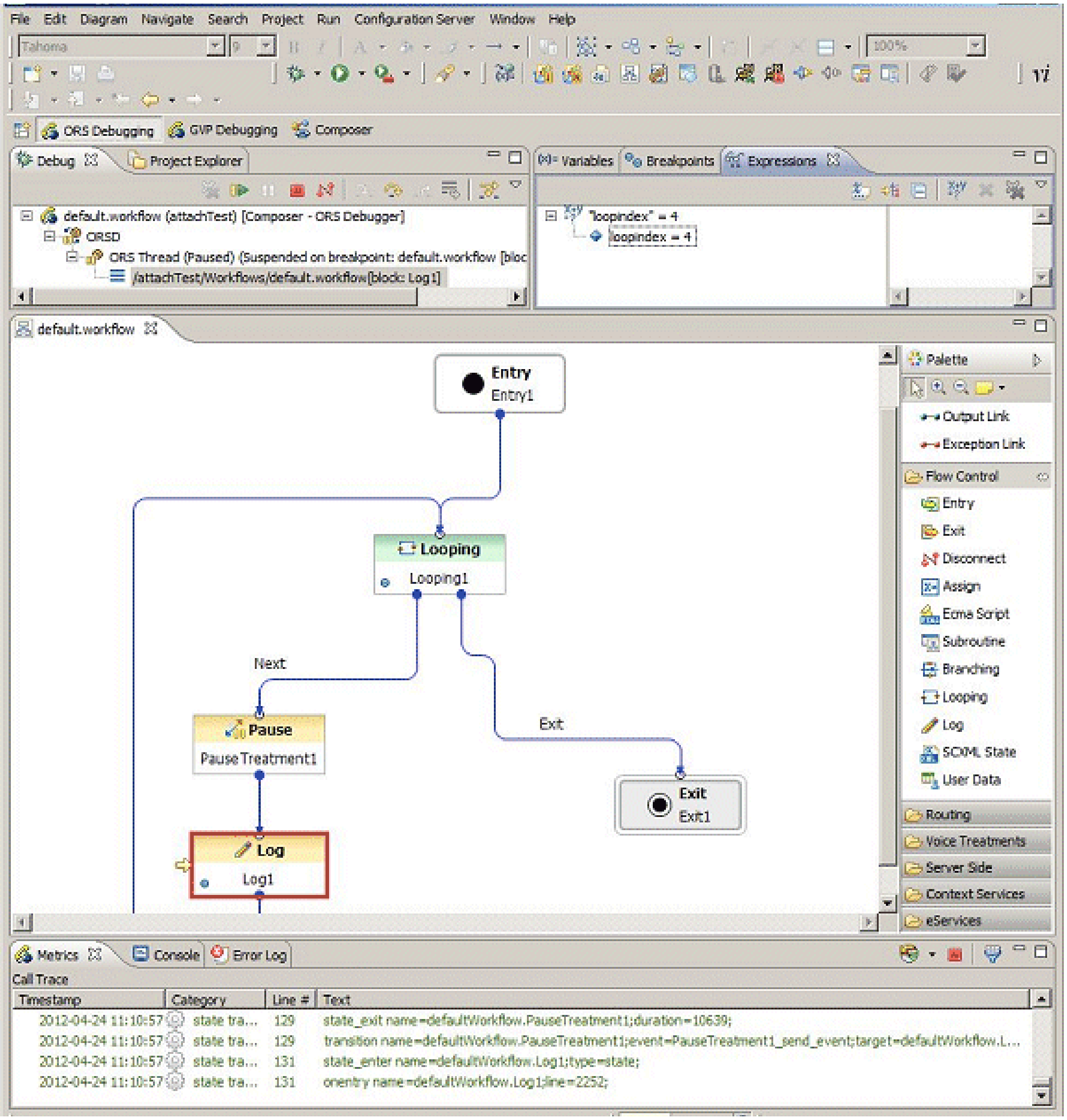
- Using a Run Configurations launch configuration, metrics (call traces) are provided and the application continues without any breakpoints. When the SCXML application executes, these metrics can describe, for example, state transitions, ECMAScript executions, and execution warnings or errors.

- 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.

Figure 7 shows a callflow in ORS Debugging perspective.



```
default.workflow (attachTest) [Composer - ORS Debugger]
  ORSD
    ORS Thread (Paused) (Suspended on breakpoint: default.workflow [bloc
      /attachTest/Workflows/default.workflow[block: Log1]
```

```
"loopindex" = 4
  loopindex = 4
```

```
Timestamp             Category      Line #  Text
2012-04-24 11:10:57   state tra...  129     state_exit name=defaultWorkflow.PauseTreatment1;duration=10639;
2012-04-24 11:10:57   state tra...  129     transition name=defaultWorkflow.PauseTreatment1;event=PauseTreatment1_send_event;target=defaultWorkflow.L...
2012-04-24 11:10:57   state tra...  131     state_enter name=defaultWorkflow.Log1;type=state;
2012-04-24 11:10:57   state tra...  131     onentry name=defaultWorkflow.Log1;line=2252;
```

Figure 7: ORS Debugging Perspective

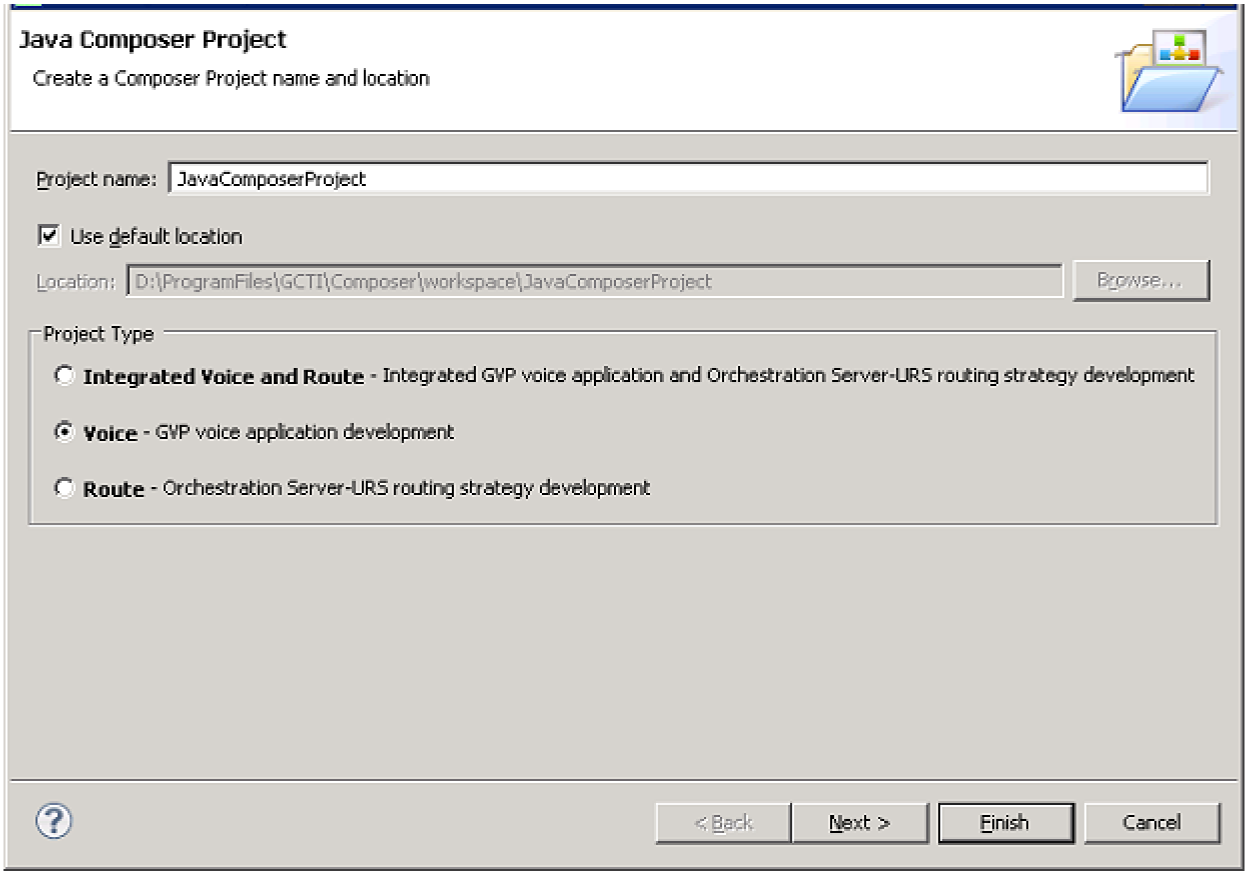
Other Composer Features

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. As shown in Figure 8, 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.



Java Composer Project

Create a Composer Project name and location

Project name: JavaComposerProject

☑ Use default location

Location: D:\ProgramFiles\GCTI\Composer\workspace\JavaComposerProject

Project Type

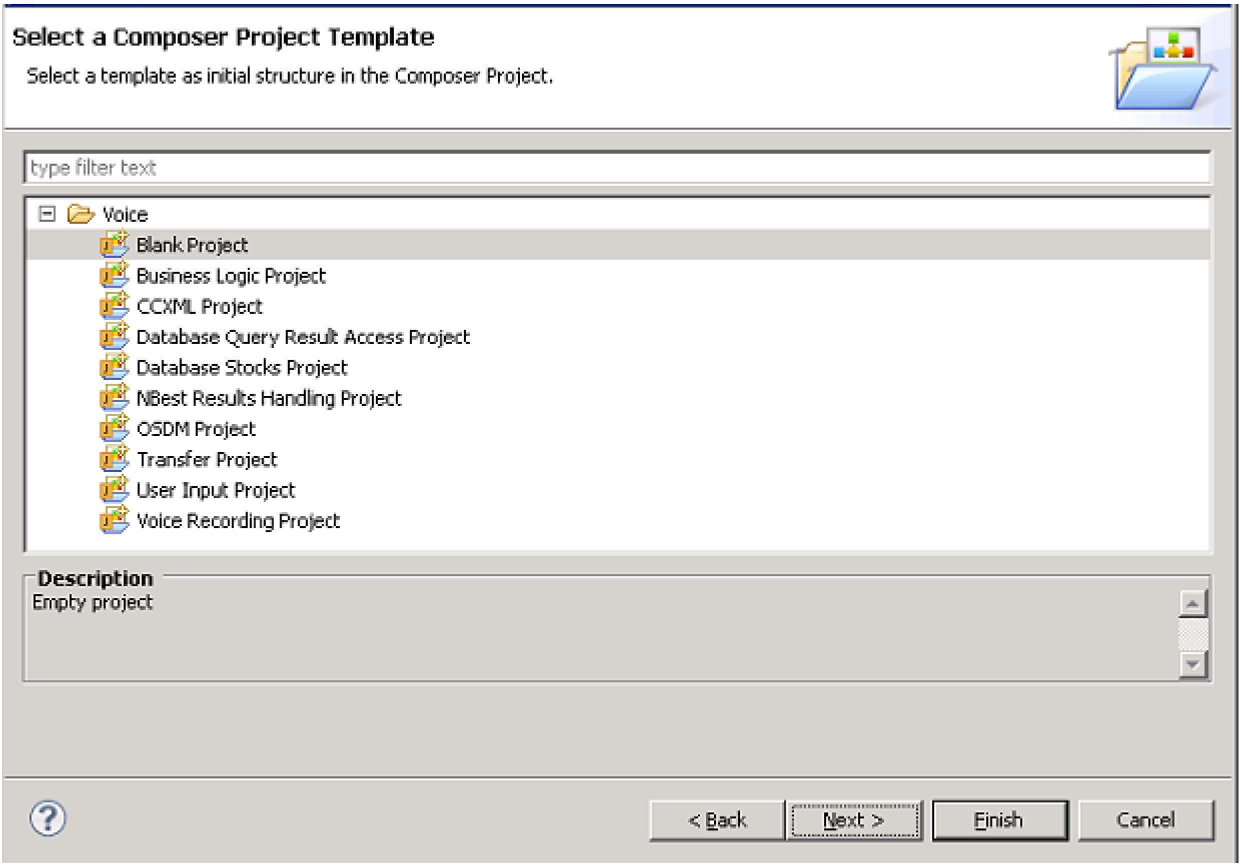
○ **Integrated Voice and Route** - Integrated GVP voice application and Orchestration Server-URS routing strategy development

● **Voice** - GVP voice application development

○ **Route** - Orchestration Server-URS routing strategy development

Figure 8: Java Composer Project Wizard

Clicking `Next` brings up available templates for the selected category (see Figure 9).



Select a Composer Project Template

Select a template as initial structure in the Composer Project.

type filter text

- Voice
  - Blank Project
  - Business Logic Project
  - CCXML Project
  - Database Query Result Access Project
  - Database Stocks Project
  - NBest Results Handling Project
  - OSDM Project
  - Transfer Project
  - User Input Project
  - Voice Recording Project

Description

Empty project

Figure 9: Voice Project Templates

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 (see Figure 1 on page 15) 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. For more information, see "Hiding Capabilities" on page 65.

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

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.

New Features in 8.1

This section describes the new Composer 8.1 features.

Composer 8.1.2

This section describes the new features in Composer 8.1.2.

Common Features Across Applications:

- The Business Rules block works directly with Genesys Rules Engine and does not require going through GRAT server at runtime. This simplifies the usage of Genesys Business rules in VXML and SCXML applications.
- Enhancements made to Database blocks support Database clusters and secure connections; this enables users to connect to Oracle RAC and SQL clusters.
- Database Connections can use service names in addition to SIDs. Connection strings can be dynamically generated and support variables. This helps developers to simplify the usage of database connections.
- Blocks in either Orchestration workflows (SCXML) or Voice call-flows (VXML) can be disabled. For example, you may wish to temporarily remove a block during debugging or during development. Disabled blocks do not participate in the application at runtime.
- New properties for Logging are available for all blocks. Additional support for Alarms is added to workflows. This feature allows developers to minimize insertion of Log blocks and improves readability.

GVP application (VXML) Enhancements:

- Support for Outbound Campaigns in callflows. New Outbound blocks enable callflow applications to update, add or delete records in Campaign Calling Lists and work as a solution in tandem with Genesys Outbound Solutions. Users can also update Do Not Call lists in an Outbound Solution through callflow diagrams.

- Callflow applications (VXML) can use the Operation Parameter Module (OPM) and Audio Resource Management (ARM) features of Genesys Administrator Extension. OPM enables simplification of the overall solution by allowing business users to easily control and manage callflows.
- This release adds a new utility function to access SIP header values in callflows.
- A new VXML code block allows the embedding of VXML code directly into callflows through <subdialog>. This feature provides developers the flexibility to modularize callflow diagrams.
- Users can specify custom formats for Voice prompts in a VXML applications. Custom formats can be created via ECMAScript functions in callflows.
- Input and ICM variables defined in callflow Entry blocks are initialized to default variables if no value is supplied at runtime. This behavior is controlled by a flag. Older version of callflows will continue to have this flag reset to maintain backward compatibility.

Orchestration Application (SCXML) Enhancements:

- Enhanced debugger support provides the ability to debug SCXML applications. The Composer interface provides full debugging functionality for Composer-generated and hand-coded applications.
- New Outbound Campaign blocks support integration with Genesys Outbound Contact features, such as adding, deleting, and updating Calling List records; updating Do Not Call lists; and other Calling List manipulation features. This functionality provides more robust integration between Routing logic and Genesys Outbound Contact functionality.
- New blocks support the SCXML <parallel> functionality allowing developers to define applications that can simultaneously perform multiple operations. Entry, Subroutine and Begin parallel blocks in workflows and sub-workflows support target-less transitions, which could be based on some condition.
- Support for Genesys Administrator Extension Operation Parameters (OPM) and Audio Resource Management (ARM) functionality in SCXML applications. This feature simpifies the solution and provides control to the end user, addressing Total Cost of Ownership (TCO).
- Voice Treatment blocks provide direct access to Extension data returned after treatment completion. Composer now supports Orchestration Server-based treatments instead of Universal Routing Server-based treatments.
- Support for multiple views for Workbins and existing queues within interaction process diagrams (IPDs).
- The Composer Help is available on the Composer Documentation Wiki.

Composer 8.1.1

This section describes the new features in Composer 8.1.1.

New Routing Application Features

New features for creating SCXML-based routing applications include:

- An Orchestration Server (ORS) Debugger, which gives ability to debug SCXML applications including routing applications. The applications can be Composer-generated, hand-coded or a mix of both.
- When specifying ORS preferences, you can enable secure communications (SSL/TLS) between the Composer client and ORS, for SCXML debugging sessions. The connection between Composer and ORS is mutually-authenticated TLS if implemented on the ORS side.
- Routing blocks, as well as those involved in interaction processing, support multi-site routing: Target, Route Interaction, Queue Interaction, Force Route, Routing Rule, Default Route, Create E-mail, E-mail Response, E-mail Forward, Chat Transcript, and Create SMS. See new properties `Detach` and `Detach Timeout`.
- Support for development of "interaction-less" processing has been added, which allows the creation of SCXML applications that may be started/interacted with via ORS Web Services, rather than an interaction. The following features support "interaction-less" processing:
    - Blocks that influence interactions now support selecting the interaction they should use. The default behavior is to use the current interaction which is backwards compatible.
    - `Wait for Event` in the interaction process diagram, which can be set to not wait for a startup or triggering event thereby enabling interaction less workflows.

  To support "interaction-less processing," the following blocks add a new property, `Interaction ID`:

    - Routing blocks: Default Route, Force Route, Route Interaction, Queue Interaction, Routing Rule, Stop Interaction, and Target.
    - Flow Control blocks: Disconnect, and Exit
    - eServices blocks: Chat Transcript, Create Email, Create SMS, Email Forward, Email Response, Identify Contact, Update Contact, Create Interaction, and Render Message.
    - Voice Treatment: Create User Announcement, Delete User Announcement, IVR, Play Application, Play Message, and Play Sound.
- Interaction process diagrams add a `Namespaces` property, which gives the ability to refer to custom namespaces in generated code.

- Interaction Queue blocks in interaction process diagrams support segmentation based on views. Multiple views can be defined and each can redirect flow to a different workflow diagram.
- The following Flow Control blocks are available when creating an interaction process diagram: Branching, ECMAScript, and Log.
- When segmenting interactions to take different paths in a workflow, you now have the ability to define a default limit for each segment.
- When using the Media Server block to specify interactions of a particular media type for an interaction process diagram, the following servers are now available for selection: Chat Servers and Third Party Servers (such as one used for Capture Point application). The Publish operation creates endpoints for these server types.
- When using the Route Interaction block, a new Hints property allows you to specify extension data. The following blocks also add the Hints property: Cancel Call, Create User Annoucement, Delete User Annoucement, Default Route, Queue Interaction, Play Application, Play Sound, Play Message, Routing Rule, Target, User Input.

- When using the Route Interaction block, a new `Hints` property allows you to specify extension data. The following blocks also add the Hints property: Cancel Call, Create User Annoucement, Delete User Annoucement, Default Route, Queue Interaction, Play Application, Play Sound, Play Message, Routing Rule, Target, User Input.

- The Play Application block adds a new property, `Use User Data`. When set to true, Composer will automatically update the interaction’s user data with the input/inout parameters specified in the `Parameters` property.
- The Target and Force Route blocks add a `Type` property, which you can use to define the type of redirection processing.
- The Route Interaction block and Target blocks add a new property, `Include Requests From Previous Blocks`, which can be used for cascaded target lookups.
- A new Wait block can be used to have Orchesration Server transition out when one of a defined list of events is received and the associated condition is true.
- When using Composer's Business Rule block to request the Genesys Rules Engine to execute a Rule Package in a routing workflow or voice callflow, the `getUData()` function is now available.

New Voice Application Features

New features for creating voice applications for GVP include:

- VXML callflows now support a VXML application root document. This enables features like global variables that are available across all callflows and sub-callflows. .

- The Prompts property in the following blocks allows VoiceXML to overlay text into an existing video image/stream: Prompt, Menu, Input, DB Prompt, DB Input, Grammar Menu, Record, and Menu.
- The Menu block supports specifying DTMF for repeating a menu.

New Voice & Route Application Features

- While exporting a .WAR file, each Project can specify a unique name which is included in the .WAR file.
- When using Context Services, you can specify a particular media type for a service, which can be a Configuration Server Business Attribute, such as for an Application Type. The following blocks add the Media Type property: Start Service, Associate Service, Complete Service, Enter State, Complete State, Start Task, and Complete Task.
- When defining parameters for the Backend, Web Request, Web Service, Subroutine, and Subdialog blocks, you can now use Expression Builder.

Composer 8.1

This section describes the new Composer 8.1 features.

IRD to Composer Migration Support

Starting with Composer 8.1, you can migrate routing strategies created with Interaction Routing Designer (IRD) 8.0+ into Composer Projects as SCXML-based workflow diagrams, which can run on the Orchestration Platform. The migration process uses an import wizard to handle the transformation from an IRD strategy into a Composer workflow diagram. The *IRD To Composer Migration Guide*, available on the on the Composer Documentation Wiki, details the migration process.

Interface with Genesys Rules Engine

Composer can now interface with the Genesys Rules Engine, which is part of the Genesys Rules System. A Composer-compatible plug-in is available for developing business Rule Templates. This plug-in is provided as part of the Genesys Rules System. For information on installing the plugin, see “Installing the Business Rules Plugin” on page 69 of this guide.

A new Business Rule block lets you request the Genesys Rules Engine to execute a particular set of business rules in a routing workflow or voice callflow and get the results back.

Note: In the Genesys 8.1 release, the Genesys Rules System will only be packaged with the intelligent Workload Distribution product and the Conversation Manager product.

New Blocks for IRD Parity

- Composer moves closer to parity with Universal Routing's strategy creation tool, Interaction Routing Designer (IRD).
- An E-mail Response block combines the functionality of IRD’s Acknowledgement, Autoresponse, and Create Notification objects.
- A Chat Transcript block allows you to generate a reply e-mail to a chat interaction and attaches a chat transcript.
- An E-mail Forward block combines the functionality of IRD's Forward E-mail, Redirect E-mail, and Reply E-mail from External Resource object.
- A Screen Interaction block allows you to screen a text-based interaction for specific content (specific words or patterns), and then (optionally) segment the interaction to different logical branches based on the result of the screening query.
- A Classify Interaction block allows you to classify a text-based interaction based on content, and attach one or more Classification categories to the interaction.
- For classification segmentation, an ECMAScript function determines if a particular category name or ID exists in the array of category objects represented by an application variable. This variable can be the output of the Classify Interaction block, enabling the Branching block to be used for segmentation based on category.
- For manually attaching categories to an interaction, the User Data block can be used and then a branching block can be (optionally) used to segment interactions to different logical branches based on the different categories.
- An Update Contact block allows you to update customer profile information in the UCS Database, based on data attached to an interaction.
- An Identify Contact block can identify a contact based on the interaction User Data; return a list of matching Contact IDs based on the User Data; create a contact record in the UCS Database with information in the User Data if a matching contact is not found; or update the UCS Database record of the matching contact with information from the current interaction’s User Data.

- A Create Interaction block allows you to create an interaction record in the Universal Contact Server Database for a customer contact. This saves the current interaction being processed by the strategy, in the database.
- A Render Message block provides the ability to render field codes in arbitrary text.

Other New Routing Application Features

New Routing Application features are as follows:

- Composer's existing Create E-mail block is enhanced to allow you to: pick up standard response text from User Data; specify that the “To” address be picked up from the Customer Profile in the Universal Contact Server Database; and use Field Codes in standard responses that will later be filled in with user-specific values.
- Composer's existing Route Interaction block now allows you to create applications where routing is based on schedules from Genesys Workforce Management.
- The Flow Control palette for routing applications contains a new SCXML State block. When used in a workflow diagram, it allows you to write custom SCXML code that Composer will include in the SCXML document that it generates based on the workflow diagram.
- The Flow Control palette for routing applications contains a new User Data block for updating an interaction's User Data and for attaching `Business Attributes`, `Categories`, and `Skills`.
- When an interaction process diagram (IPD) uses a Workflow block, if the referenced workflow diagram contains an eServices block that names a server performing an action or operation, Composer adds a visual indicator in the form of a node (similar to an IRD strategy-linked node).
- When developers work with Context Services, Composer accepts HTTP basic authentication credentials and uses them for authentication, including digest authentication for working with Web Services.
- You can now use variables in Skill Expression Builder. You can also disable Skill Expression validation from the Configuration Server preference page.
- You can now include your own custom JavaScript (`*.js`) files in workflows by placing them in the `/include/user` folder. The JavaScript functions in the specified `.js` file can then be used in Assign or Branching block expressions.
- A new Composer Route Project template is available: `Forward to External Resource`.
- Composer’s database Query Builder and Stored Procedure Helper now support table synonyms.

- New Integrated Voice and Route Project templates are available: `Load Balancing and Working Hour Routing`, `External File-Based Routing`, and `Play Application and Busy Treatment`.

GVP-Specific Enhancements

To support creating voice applications for Genesys Voice Portal (GVP):

- The Transfer block provides a property for setting an authorization code (`authcode`). It can be populated either from free-form text or from an application variable.
- The Call Trace view used for debugging a callflow displays the line number for each incoming metric.
- A "barge-in" option is available for prompts. The Interruptible property for the following blocks add a new option for DTMF-only barge-in mode: Prompt block, DB Prompt block, Input block, Menu block, Grammar Menu block, and DB Input block.
- Automatic selection of language-specific pre-recorded prompts, grammars, and TTS prompts is now available during application execution. The following blocks add a new Language property: Prompt block, DB Prompt block, Input block, Menu block, Grammar Menu block, DB Input block, and Record block.

  The language set by this property overrides any language set by the Set Language block, the Project preferences, or the incoming call parameters. The property takes effect only for the duration of the block. The language setting reverts back to its previous state after the block is done.
- The Language property affects the language of grammars used for ASR input for the following blocks: Input block, Transfer block, and Route Request block.
- The Record block's Capture Filename Prefix and Capture Location properties now allow selection from application variables in addition to accepting literal strings.
- You can now use the GVP ICM Adapter in VoiceXML applications, including invoking services, responding to requests, and sharing data.
- A new ICM Interaction Data block, available on the CTI Blocks palette, supports sending of variables to Intelligent Contact Management (ICM).
- A new ICM Route Request block, also available on the CTI Blocks palette, supports routing the call to CTI.
- The Exit block's Return Values property dialog allows you to select the ICM variables to be returned.
- Voice Projects now have a Project-level flag, which controls whether ICM variables are available for selection and assignment to variables within Composer's Entry block.
- SSML tags can be used in prompts.

Security Enhancements

This release includes the following security-related enhancements:

- The Web Service block now supports certificate-based authentication. You can develop both voice (VXML) and routing (SCXML) applications that support secure mutual authentication and communication with a Web Service. Composer supports the use of both a digital client certificate and server certificate contained in a keystore file.
- When creating a routing application and connecting to Configuration Server, Composer displays informational text associated with both successful and unsuccessful authentication.
- You can configure an inactivity timeout for the connection to Configuration Server as well as when the timeout warning dialog should appear.
- You have the option of having a configurable security banner appear when Composer is first launched, similar to other Genesys applications. For information on configurable items related to the banner, see the *Genesys 8.1 Security Deployment Guide*.
- Composer supports secure connections when connecting to GVP's Media Control Platform and when connecting to Context Services and for Universal Contact Server.
- Composer now has Transport Layer Security (TLS) support and adheres to Federal Information Processing Standards (FIPS) in its connection to Configuration Server and to GVP Media Control Platform.

Routing & Voice Applications

When organizing custom blocks, you have the option of creating new drawers in the palette. You can also select from a set of bundled custom icons for the custom blocks you create.

Expression Builder is enhanced as follows:

- It now returns to its last state when re-opened, which includes displaying the tree and the location in the tree in the Expression Builder Data area.
- The filter now works on the description of the functions in addition to the function signatures.
- Data loading is optimized to run in a separate thread. As a result, dialogs remain responsive while data loading is in progress.

New operating system support for 8.1 is as follows:

- Composer can run on the Windows 2003, Windows 2008 (32-bit and 64-bit in 32-bit compatibility mode), Windows XP, Windows Vista, and Windows 7 (32-bit and 64-bit in 32-bit compatibility mode) operating systems. For more information, see “Operating Systems Supported” on page 37.

Chapter

2 Installation

This chapter describes the Composer software requirements, and installation and launching procedures. It contains the following sections:

Software Requirements

Before you install Composer, you must consider all of the environment variables that are needed for a successful deployment.

Operating Systems Supported

Each workstation that you use to develop voice applications with Composer must meet one of the following operating system requirements:

- Microsoft Windows Server 2003 with Service Pack 2 or later
- Microsoft Windows XP with Service Pack 3 or later
- Microsoft Windows Vista
- Microsoft Windows 7, 32-bit and 64-bit in 32-bit compatibility mode.
- Microsoft Windows Server 2008, 32-bit and 64-bit in 32-bit compatibility mode.

Note: GVP 8.1 and the Composer Debugger interface both use Internet Protocol version 4 (IPv4). If you are deploying Composer on Windows Vista, make sure that you have IPv4 enabled for the network interfaces.

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.

Note: A .NET Composer Project (Voice or Routing or both) can be deployed only to Microsoft Internet Information Services (IIS, formerly called Internet Information Server). You can use any version of Microsoft IIS that is compatible with the Windows versions on which Composer is supported (see page 37).

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” on page 39.

Note: 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.

Bundled Tomcat

Composer installs an embedded Tomcat 6.0 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 6.0 as a stand-alone web server, depending upon your needs and the recommendations of your IT department. 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.

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" on page 52
- "Configuring proxy settings in Tomcat 6.0" on page 60

Internet Information Server (IIS)

See the following sections ahead:

- "Configuring in IIS Manager" on page 53
- "Configuring IIS/.NET Preferences" on page 55
- "Adding MIME Types for IIS" on page 56
- "Proxy Configurations for .NET Composer Projects" on page 60

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 (see Figure 9 on page 25) 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):

Table 1: Application Server Test Plan Project Templates

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

Note: If you are using both the voice and route features, Genesys recommends that you test both features by running two tests.

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

Composer 8.1 supports the following databases/servers:

Microsoft SQL Server 2005 and 2008.

Note: When installing SQL Server, select SQL Server authentication (Composer does not support integrated Windows authentication).

Oracle 10g, R1 and Oracle 11g.

Note: 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.

SIP Phones Supported

Composer supports the following SIP phones:

- X-Lite 3.0 (recommended if you are connecting over a virtual private network)
- Pingtel 2.4.3
- SJphone version 1.65 or later
- Express Talk 3.08 (recommended if you are using it on a Microsoft Remote Desktop connection)

Web Browsers Supported

Composer supports the following web browsers:

- Microsoft Internet Explorer 7.0 and 8.0
- Mozilla Firefox 6.0 or earlier

Third-Party Software Requirements

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

- Microsoft .NET Framework 2.0 and 3.5 (*both* are required for .NET Composer Projects)
- 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.

Genesys Software Prerequisites

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

- Orchestration Server 8.1.200.51+ for debugging and testing SCXML session-based applications.
- Genesys Voice Platform (GVP) Media Control Platform 8.1.6 for for video text overlay and Context Services authentication.
- If you wish to use Context Services in routing workflows and voice callflows, you will need Universal Contact Server/Context Services 8.1,000,10+.
- If you wish to use the Outbound blocks, you will need Outbound Contact Server 8.1.100.09+.
- Genesys Rules System 8.1.0 or later for business logic, which can be customized, and then invoked by VXML and SCXML applications.

Note: For information on components required by Orchestration Server, such as eServices components for processing multimedia interactions, see the Packaging section of the *Orchestration Server 8.1 Deployment Guide*.

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.

Installing Composer

Before you install Composer, make sure that:

- 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 reviewed the chapter on configuring a security banner in the *Genesys 8.1 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.

Note: For 8.x releases, you can use a Microsoft Remote Desktop connection to install components. Previously, for 7.x releases, Genesys did not recommend using it.

Security Banner Configuration

Composer installation gives the option of configuring a security banner that displays when users connect to Configuration Server. The security banner appears in a separate window that is displayed to a user when logging in. 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. You specify the message content by specifying a URL pointing to a document. You may specify multiple URLs to achieve redundancy in case a URL cannot be retrieved.

Be aware of the following limitations:

- Because there is no Application object in Configuration Server for Composer. The security banner configuration is read from the registry. Make sure you select the option `All applications that support this feature` when installing Composer.
- The timeout, width, height, title registry values (and the corresponding options in the setup wizard) are ignored.
- Environment variable strings (`%SystemRoot%, ...`) are not supported in URLs.
- 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 circonstances, 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.

Procedure:
Registry Edit for Pending Operations Message

If you receive the above message after un-installing or installing Composer:

Start of procedure

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.

End of procedure

Installing Composer on Windows

See page 37 for a list of supported operating systems.

Procedure:
Installing on Windows

Start of procedure

1. Navigate to the folder containing the `setup.exe` file (`*_ENU\solution_specific\Windows\Composer`).

   Or, if you have a CD, insert it into the computer on which you want to install Composer and navigate to the folder containing the `setup.exe` file.

   Auto-run starts, and the Genesys Installation Wizard's `Welcome` screen appears.

   Note: When installing on Windows 7 or Windows Server 2008, 32-bit, ignore the "unidentified Publisher" warning which pops up.

2. Click `Next` on the `Welcome` screen. The `Security Banner Configuration` screen appears.
3. Select `Enable Security Banner` if you wish to do so at this time. For information on security banner configuration, consult the *Genesys 8.1 Security Deployment Guide*. If you do not wish to configure a security banner at this time, click `Next`.

   The `Choose Destination Location` screen appears.

   Note: Installation of Composer in a non-default location may sometimes fail if Composer has previously been installed and un-installed on the machine. The root cause is an issue with Windows Installer. As a workaround, reboot the machine and reinstall Windows Installer.

4. Accept the default installation path in the `Destination Folder` text box, or click `Browse` and select an alternate destination.

   Note: When installing Composer, do not specify parenthesis '(' or ')' in the installation path.

5. Click Next.

> **Note:** If you have voice projects from an earlier version of Composer and you choose the default folder location in this step, you still can access those projects by switching Workspaces at any time, or by importing the earlier projects to your new Workspace. For details, see "Upgrading Projects/Diagrams" on page 68.

   The Composer Parameters screen appears.

6. In the Port text box, enter the port that will be used to handle Tomcat (for example: 8082). Click Next.
7. Optional. Configure a security banner (see page 43).
8. Click Next. The Ready to Install screen appears.
9. Click Install.The Installation Status screen appears.
10. On the Installation Complete screen, click Finish or see the note below.

> **Note:** If the option to restart the machine appears, please select the option to restart now in order to have Composer properly complete the installation.

End of procedure

Launching Composer

This section gives step-by-step instructions for launching the Composer GUI.

Procedure: Launching on Windows Server 2003, XP, and Vista

Start of procedure

1. From the Windows `Start` menu, select `Programs > Genesys Solutions > Composer 8.1 > Start Composer.`

   A workspace dialog box opens to allow you to select the location for your project files.

2. Accept the default entry, or click `Browse` to navigate to a location that will serve as your workspace folder.

   Note: For Windows Vista, please be sure that your workspace folders will be *outside of* the `Program Files` folder. When prompted for a workspace folder, do not specify parenthesis '(' or ')' in the workspace path.

3. If you want your selected workspace to be your default and do not want to select a location the next time that Composer opens, select the `Use this as the default and do not ask again` check box.

4. Click `OK` to proceed.

   The first time you launch Composer, a `Welcome` tab appears. At this point, you can:

   - Get an overview of the features.
   - Go through tutorials.
   - Read more on the web.
   - Go to the workbench.

5. If you choose not to explore those links at this time, click the `X` on the `Welcome` tab to close it and display the Composer GUI as shown in the Overview chapter of this guide. You can always access the Welcome screen at any time from within Composer by selecting `Help` > `Welcome`.

End of procedure

At this point, you may find it helpful to load a Project template so you can see a finished application. Follow the procedure in the next section if you wish to do this.

Procedure: Launching on Windows 7 and Windows Server 2008, 32-bit operating systems

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

Start of procedure

1. Run / launch Composer 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`

End of procedure

Viewing a Sample Application

Composer provides a set of predefined Project templates (see Figure 9 on page 25) 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.

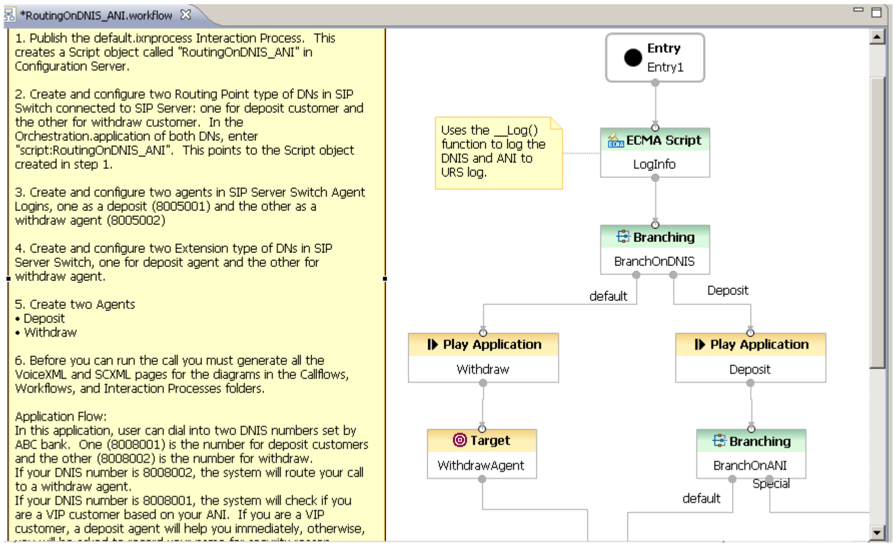
Procedure: Sample integrated voice and route application

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

Start of procedure

1. Select `File` > `New` > `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`. Your selection will determine which Project templates are shown in the next dialog box. For this example, select `Integrated Voice and Route`.
4. Click `Next`. The `Select a Composer Project Template` dialog box opens.

5. Select the `Routing Based on DNIS and ANI` Project template and click `Finish`. A commented `RoutingOnDNISandANI.workflow` diagram appears on the canvas (see Figure 10).



*RoutingOnDNIS_ANI.workflow

1. Publish the default.ixnprocess Interaction Process. This creates a Script object called "RoutingOnDNIS_ANI" in Configuration Server.

2. Create and configure two Routing Point type of DNs in SIP Switch connected to SIP Server: one for deposit customer and the other for withdraw customer. In the Orchestration.application of both DNs, enter "script:RoutingOnDNIS_ANI". This points to the Script object created in step 1.

3. Create and configure two agents in SIP Server Switch Agent Logins, one as a deposit (8005001) and the other as a withdraw agent (8005002)

4. Create and configure two Extension type of DNs in SIP Server Switch, one for deposit agent and the other for withdraw agent.

5. Create two Agents
• Deposit
• Withdraw

6. Before you can run the call you must generate all the VoiceXML and SCXML pages for the diagrams in the Callflows, Workflows, and Interaction Processes folders.

Application Flow:
In this application, user can dial into two DNIS numbers set by ABC bank. One (8008001) is the number for deposit customers and the other (8008002) is the number for withdraw.
If your DNIS number is 8008002, the system will route your call to a withdraw agent.
If your DNIS number is 8008001, the system will check if you are a VIP customer based on your ANI. If you are a VIP customer, a deposit agent will help you immediately, otherwise,

Uses the __Log() function to log the DNIS and ANI to URS log.

Figure 10: Sample Application Integrated Voice and Route Application

6. Double-clicking a block opens its `Properties` view.

End of procedure

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

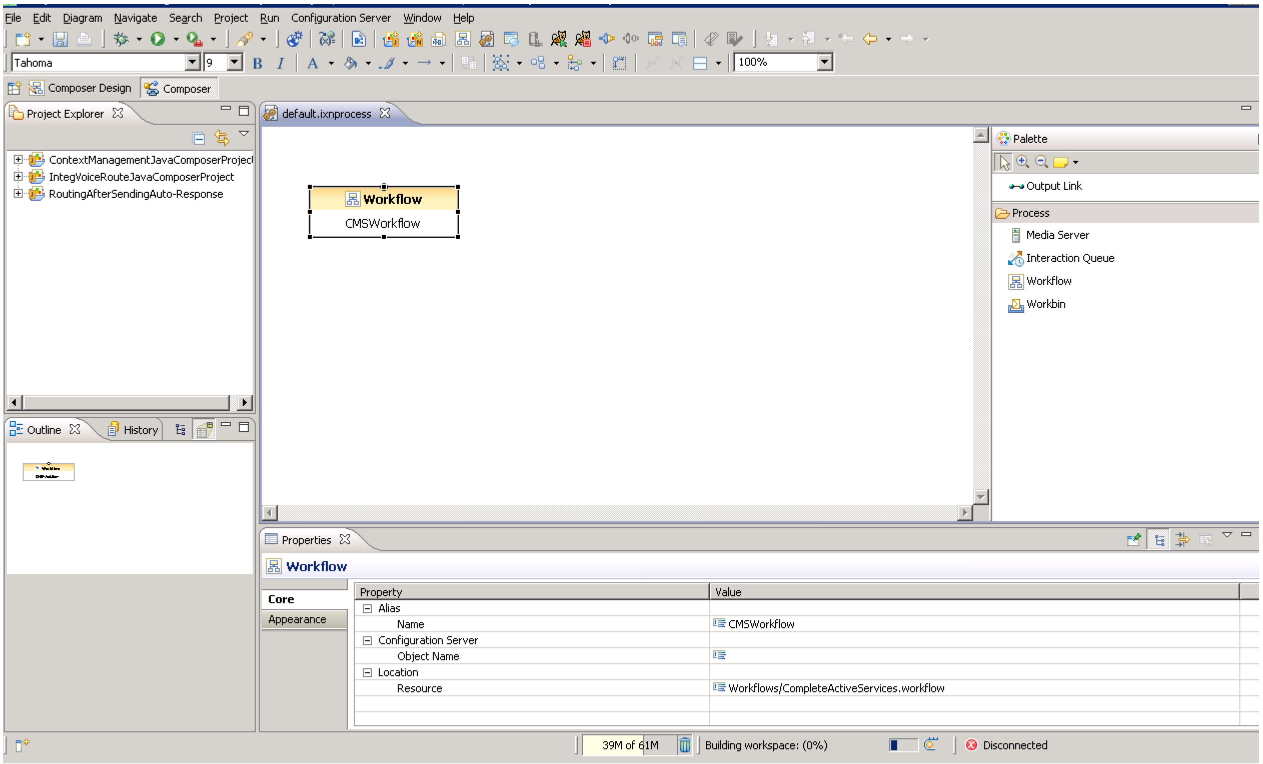
Procedure:
Sample routing application

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

Start of procedure

1. Select `File` > `New` > `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 (see Figure 11).



Property	Value
Alias	
Name	CMSWorkflow
Configuration Server	
Object Name	
Location	
Resource	Workflows/CompleteActiveServices.workflow

Figure 11: Sample Routing Application

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 `Project 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.

End of procedure

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

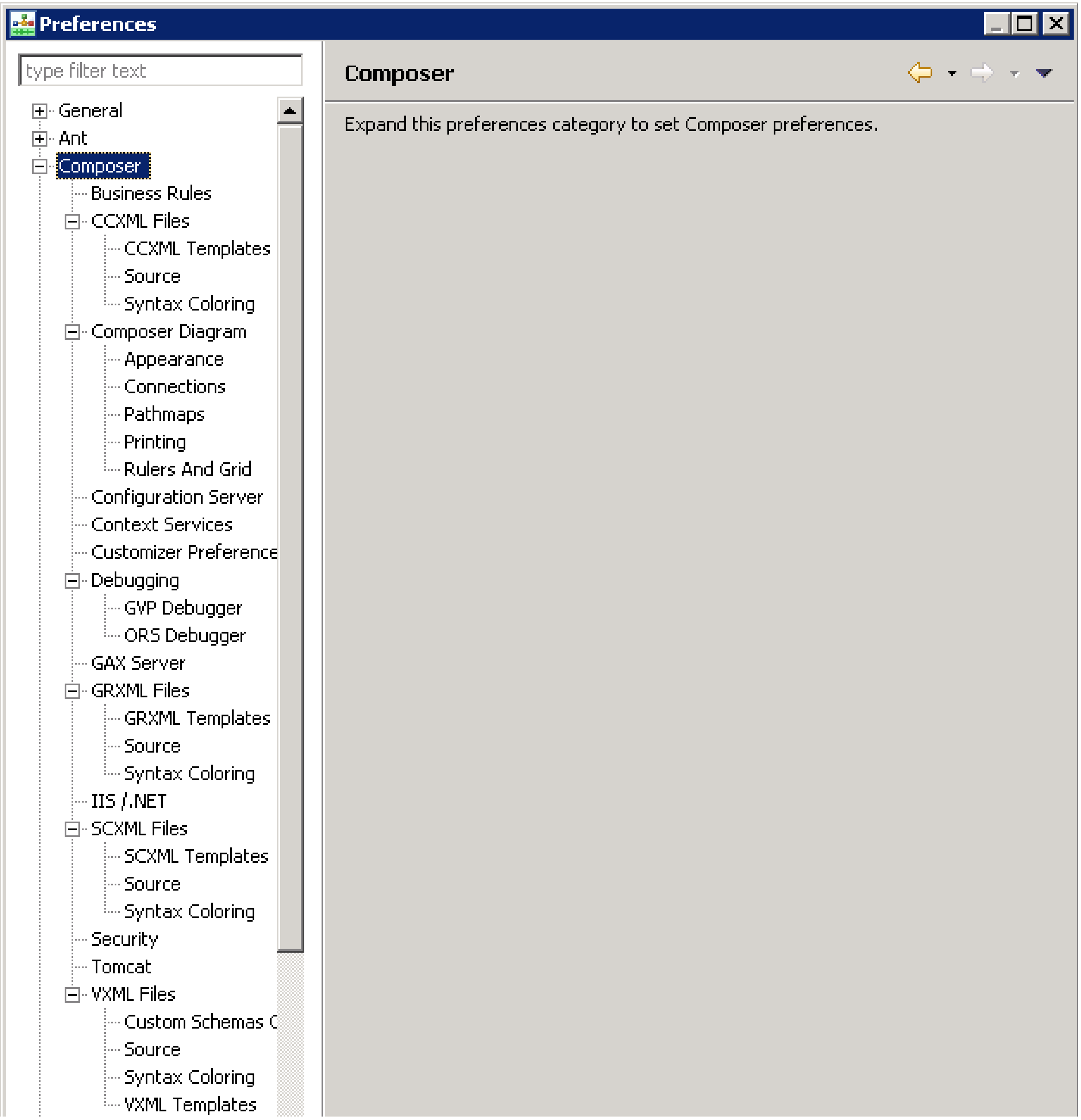
Chapter

3 Post Installation Configuration

This chapter contains post-installation tasks to be performed after installing and launching Composer. It contains the following sections:

Configuring Tomcat Settings

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`). Figure 12 shows the Composer Preferences dialog box fully expanded.



Expand this preferences category to set Composer preferences.

Figure 12: Composer Preferences

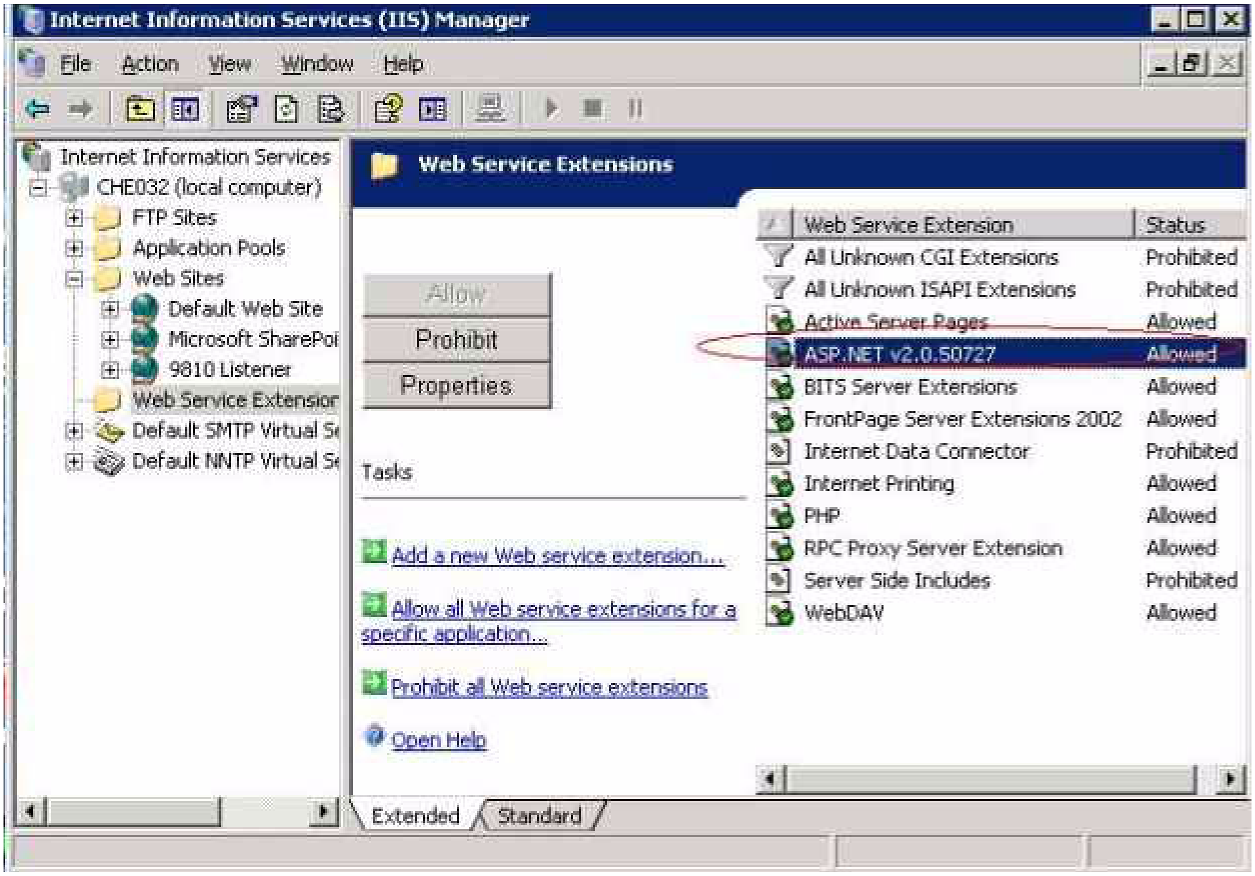
For more information, see the Tomcat section in the Voice Applications and Callflows book, Callflow Post Installation Configuration page or Routing

Applications and Workflows book, Workflow Post Installation Configuration page, on the Composer 8.1 Help Wiki.

Configuring in IIS Manager

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 settings for Composer (see page 55).
- Allow the ASP.NET Web Service extension in Internet Information Services (IIS) Manager (see Figure 13).

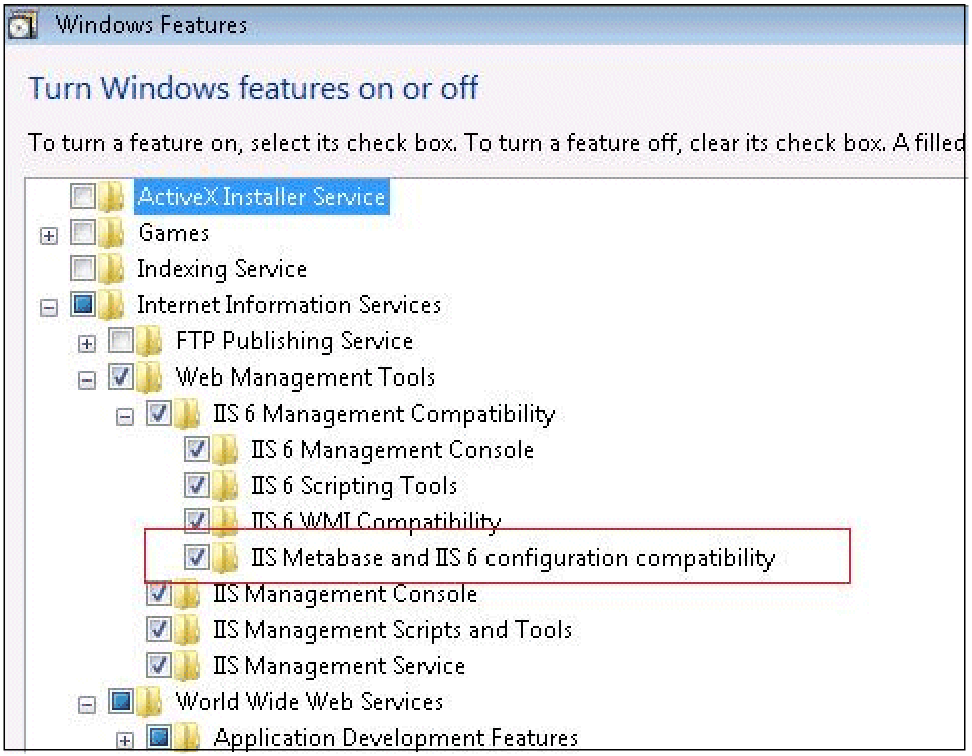


Web Service Extension	Status
All Unknown CGI Extensions	Prohibited
All Unknown ISAPI Extensions	Prohibited
Active Server Pages	Allowed
ASP.NET v2.0.50727	Allowed
BITS Server Extensions	Allowed
FrontPage Server Extensions 2002	Allowed
Internet Data Connector	Prohibited
Internet Printing	Allowed
PHP	Allowed
RPC Proxy Server Extension	Allowed
Server Side Includes	Prohibited
WebDAV	Allowed

Figure 13: Allow ASP.NET Web Service Extensions in IIS Manager

Notes: 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, IIS Metabase and IIS 6 configuration compatibility feature must be installed.Select `Control Panel` > `Programs and Features` > `Turn Windows features on or off` > `Roles` > `Add Role Services` (see Figure 14).



Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled

- ☐ ActiveX Installer Service
- ☐ Games
- ☐ Indexing Service
- ■ Internet Information Services
  - ☐ FTP Publishing Service
  - ☑ Web Management Tools
    - ☑ IIS 6 Management Compatibility
      - ☑ IIS 6 Management Console
      - ☑ IIS 6 Scripting Tools
      - ☑ IIS 6 WMI Compatibility
      - ☑ IIS Metabase and IIS 6 configuration compatibility
    - ☑ IIS Management Console
    - ☑ IIS Management Scripts and Tools
    - ☑ IIS Management Service
  - ■ World Wide Web Services
    - ■ Application Development Features

Figure 14: IIS Metabase and IIS 6 Configuration Compatibility

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

Procedure: Allowing ASP.NET Web Service Extensions in IIS

Start of procedure

1. Open Microsoft Internet Information Services (IIS) Manager on your computer.

   The `Internet Information Services (IIS) Manager` window opens as shown in Figure 13 on page 53.
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.

End of procedure

Configuring 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 (see Figure 12 on page 52) 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 information on configuring Tomcat, see the Voice Applications and Callflows book, Callflow Post Installation Configuration page, Internet Information Services section, on the `Composer 8.1 Help Wiki`.

Note: 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.

Adding MIME Types for IIS

This procedure is only necessary if you are using the Internet Information Services (IIS) Application Server to deploy ASP.NET projects.

Multipurpose Internet Mail Extensions (MIME) settings already are pre-configured in the Tomcat 6.0 web server that is bundled with Composer. MIME Types must be added *manually* for the Microsoft IIS web server.

Procedure: Adding MIME Types in IIS

Start of procedure

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).

End of procedure

Configuring the GVP Debugger

In order to make test calls, you must configure GVP Debugger Preferences (see Figure 12 on page 52). . The GVP Debugger 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 information on configuring the GVP Debugger, see the Callflow Post Installation Configuration page, GVP Debugger section, in the.

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 (see Figure 12 on page 52).

For more information, see the Routing Applications and Workflows book, Workflow Post Installation Configuration page, ORS Debugger section, on the `Composer 8.1 Help Wiki`..

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 (see Figure 12 on page 52).

For information on enabling debugging in MCP, see the Voice Applications and Callflows book, Callflow Post Installation Configuration page, Media Control Platform section, on the `Composer 8.1 Help Wiki`.

Configuring TCP Ports

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 (see Figure 12 on page 52).

For more information, see the Voice Applications and Callflows book, Callflow Post Installation Configuration page, Firewall section, on the Composer 8.1 Help Wiki.

Configuring 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.

Set GAX Preferences (see Figure 12 on page 52) if you plan on using the OPM (Operational Parameter Management) 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 the Routing Applications and Workflows book, Prefernces for Routing Applications, Gax Server Preferences page, on the Composer 8.1 Help Wiki.

Configuring 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 Figure 12 on page 52).

For more information, see the Common Blocks and Funtionality book, Business Rule Common Block page, Business Rules Preferenes section, on the Composer 8.1 Help Wiki.

Configuring Proxy Settings

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

Local Proxy

Procure:
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.

Note: Configure proxy settings only if you are accessing a URL that is *outside* of your network, and you are using a `Web Service` or `Web Request` block.

Start of procedure

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

End of procedure

Next Steps

- You must also configure proxy settings for the bundled Tomcat 6.0 web server.

Proxy Settings—Tomcat 6.0

Procedure:
Configuring proxy settings in Tomcat 6.0

Proxy settings have to be configured in the bundled Tomcat 6.0 web server for the back-end pages to access the Web if you plan to use the Web Request and Web Service blocks.

Start of procedure

1. To configure proxy settings in the bundled Tomcat 6.0 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
```

Note: 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: `Composer80Tomcat`.

End of procedure

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.

Procedure:
Configuring proxies for .NET Composer Projects

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.

Start of procedure

To configure proxy settings in a .NET Composer Project:

1. Open the `web.config` file.
2. Go to the <system.net> section.

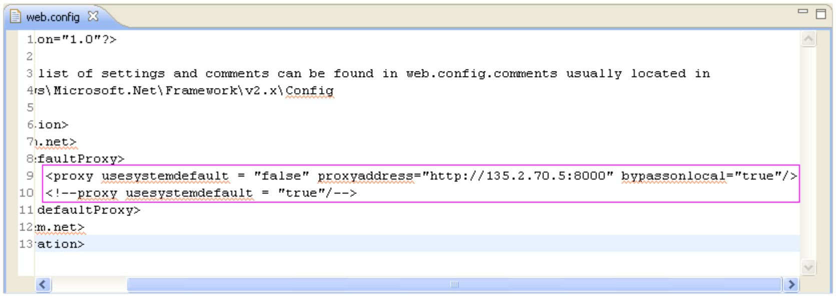
3. To use the default System proxy settings: `<proxy usesystemdefault="true"/>`

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

4. Set `proxy usessystemdefault` to `false`.
5. Explicitly designate the proxy server:

```
<proxy usesystemdefault="false" proxyaddress="http://address"
bypassonlocal="true"/>
```

An example is shown in Figure 15.



```
on="1.0"?>

 list of settings and comments can be found in web.config.comments usually located in
s\Microsoft.Net\Framework\v2.x\Config

ion>
).net>
faultProxy>
 <proxy usesystemdefault = "false" proxyaddress="http://135.2.70.5:8000" bypassonlocal="true"/>
 <!--proxy usesystemdefault = "true"/-->
defaultProxy>
m.net>
ation>
```

Figure 15: Example Proxy Settings in web.config

End of procedure

Prompt Resource Validation

This Composer Preference (see Figure 12 on page 52) 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. How to access diagram preferences is shown in Figure 12 on page 52.

For configuration information, see the Voice Applications and Callflows book, Callflow Post Installation Configuration page, Prompt Resource Validation section, on the `Composer 8.1 Help Wiki`.

Configuration Server Connection

When creating routing applications in Composer, you use the Configuration Database and Configuration Server.

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 information, see the Routing Applications and Workflows book, Workflow Post Installation Configuration page, Configuration Server section, on the Composer 8.1 Help Wiki.

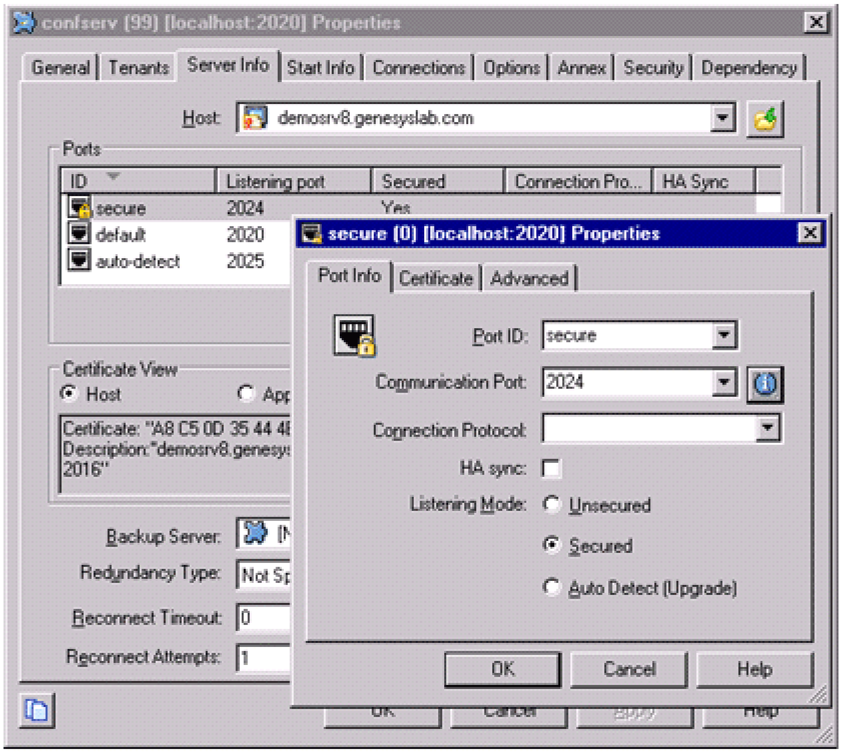
TLS support when connecting to Configuration Server

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.

Start of procedure

1. Generate and install certificates. See section “Certificate Generation and Installation” of the *Genesys 8.1 Security Deployment Guide* for instructions.

Add a secure listening port in the Configuration Server configuration. See section “Genesys TLS Configuration” of the *Genesys 8.1 Security Deployment Guide* for instructions. An example configuration is shown below (see Figure 16).



Host: demosrv8.genesyslab.com

Ports

ID	Listening port	Secured	Connection Pro...	HA Sync
secure	2024	Yes		
default	2020			
auto-detect	2025			

Certificate View: ● Host ○ App

Certificate: "A8 C5 0D 35 44 4
Description:"demosrv8.genesy
2016"

Backup Server: [N

Redundancy Type: Not Sp

Reconnect Timeout: 0

Reconnect Attempts: 1

Port ID: secure

Communication Port: 2024

Connection Protocol:

HA sync: □

Listening Mode: ○ Unsecured ● Secured ○ Auto Detect (Upgrade)

Figure 16: Configuration Server Secure Listening Port

2. 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

These Configuration Server Preferences (see Figure 12 on page 52) control routing pre-defined statistics creation, Expression Builder (see page 18) object validation, and Configuration Database object preferences. How to access Configuration Server preferences is shown in Figure 12 on page 52.

For information, see the Routing Applications and Workflows book, Preferences for Routing Applications, Configuration Server Preferences page, on the `Composer 8.1 Help Wiki`.

Inactivity Timeout

If a Composer user has authenticated with Configuration Server as described in Connecting to Configuration Server, Composer times out after a configurable number of minutes of inactivity. In this case, the user must reauthenticate in order to continue working with Configuration Server. For the 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 in the *Genesys 8.1 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 (ORS) must be properly configured.

For more information, see the *Composer 8.1 Help*.

- Open `Routing Applications & Workflows` > `Getting Started with Routing Applications`. Refer to topic Post Installation Configuration for Workflows. See section Universal Routing Tasks.

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

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

Procedure: Specifying the URL of the Starting SCXML Page

Purpose: To inform Orchestration Server of the Application Server URL.

Start of procedure

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 Deployment Guide*, SCXML Strategy Support chapter.
   - Also see chapter Configuring Other Options That Affect Orchestration Server. Refer to section Configuring the application Option on a DN, RoutePoint, or eServices Queue Object.

   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.

End of procedure

Using Stream Manager for Voice Treatments

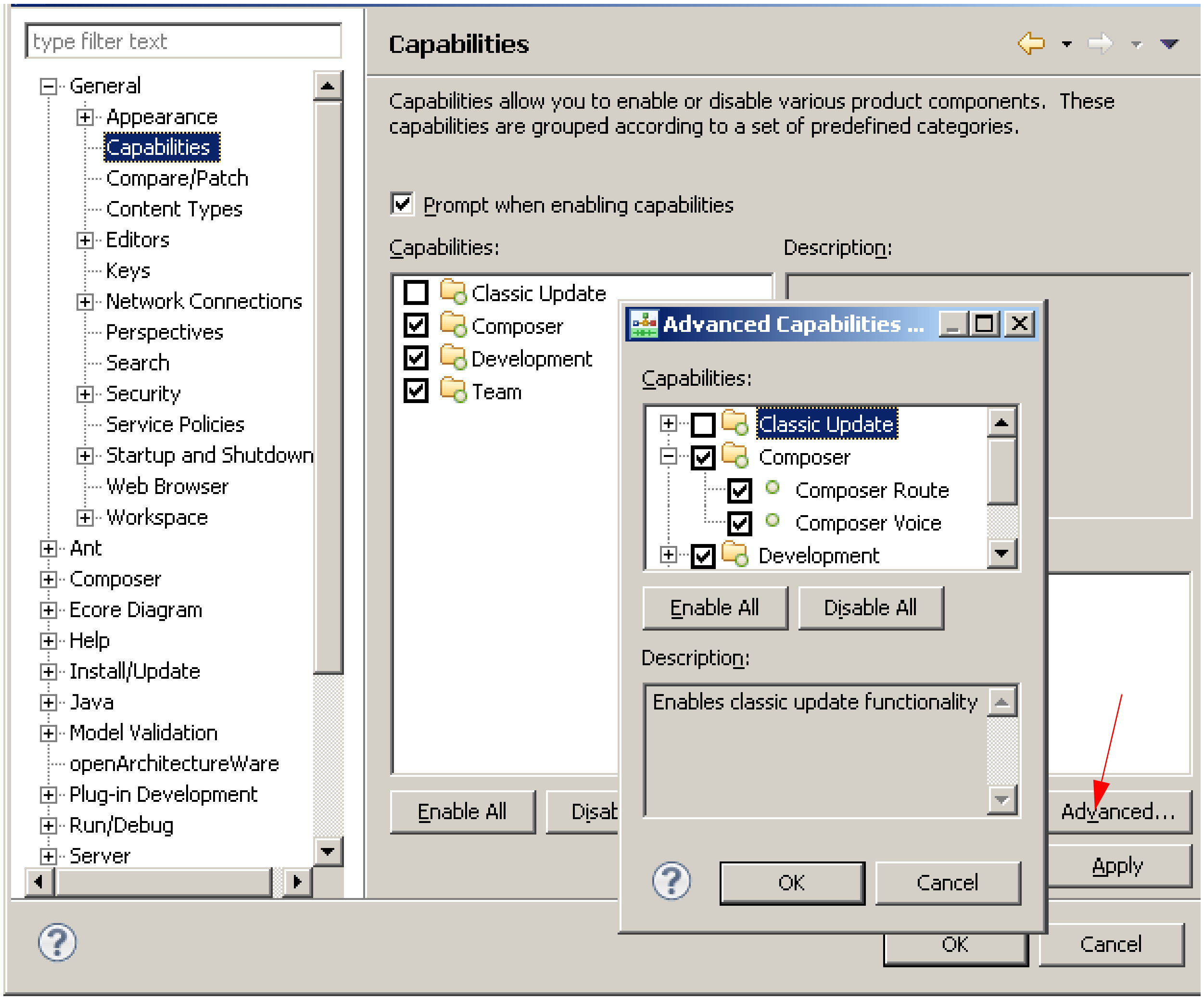
When creating routing applications, if you plan to use Stream Manager to play treatments via the Treatment blocks for workflows (such as Play Sound), additional configuration tasks are required.

For more information, see the Routing Applications and Workflows book, Workflow Post Installation Configuration page, Stream Manager section, on the `Composer 8.1 Help Wiki`.

For information on Stream Manager, start with the *Framework 7.6 Stream Manager Deployment Guide*.

Hiding Capabilities

You may hide voice or routing capabilities through a General Preference setting (`Window` > `Preferences` > `General` > `Capabilities`) (see Figure 17).



Capabilities

Capabilities allow you to enable or disable various product components. These capabilities are grouped according to a set of predefined categories.

☑ Prompt when enabling capabilities

Capabilities:

- ☐ Classic Update
- ☑ Composer
- ☑ Development
- ☑ Team

Advanced Capabilities ...

Capabilities:

- ☐ Classic Update
- ☑ Composer
  - ☑ Composer Route
  - ☑ Composer Voice
- ☑ Development

Description:

Enables classic update functionality

Figure 17: Capabilities Preferences

For more information, see the Introduction to Composer book, Enabling/Disabling Functionality and Hiding File Types pages, on the `Composer 8.1 Help Wiki`.

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 informatation, see the Common Blocks and Functionality book, Setting Context Services Preferences page, on the Composer 8.1 Help Wiki.

Procedure:
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.

Notes: 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.

Start of procedure

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

1. Generate and install a certificate in UCS. Genesys personnel may find instructions in sections "Configuring TLS" and "TLS in Context Services - HTTPS" of the CMS Design TLS UCS documentation wiki. (`http://usnavstechpubs01p.ndc.alcatel-lucent.com/dev_teams/index.php/CMS_Design_TLS#TLS_in_Context_Services_-_HTTPS`).
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 available at `http://download.oracle.com/javase/1.4.2/docs/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 Serivces` preference page, enable the TLS by selecting `TLS` in the `Security Settings` area.

End of procedure

Procedure: Using Context Services authentication during design

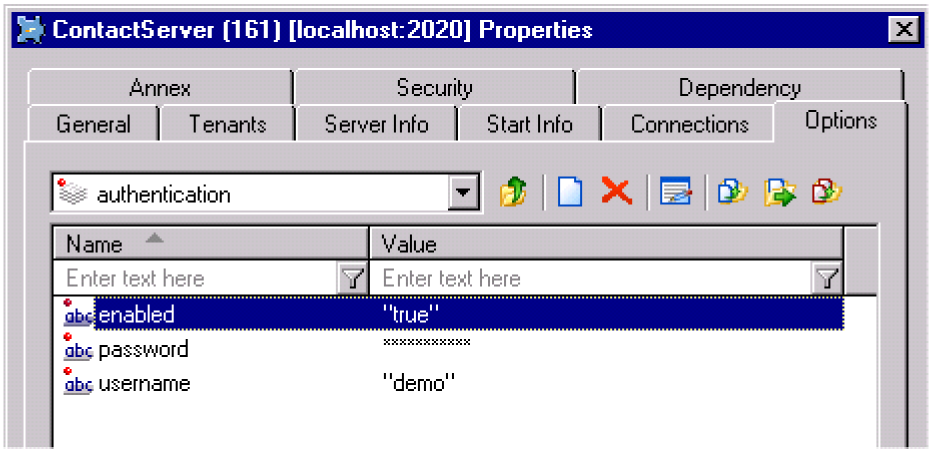
This section summarizes how to configure Composer to connect to Universal Contact Server using authentication when designing routing applications.

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

Start of procedure

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

1. Set the authentication mode to true in the UCS configuration. An example is shown below (see Figure 18).



Name	Value
enabled	"true"
password	xxxxxxxxxxxx
username	"demo"

Figure 18: Universal Contact Server Application, authentication section

See the UCS Team documentation wiki for details (`http://usnavstechpubs01p.ndc.alcatel-lucent.com/dev_teams/index.php/CMS_Design_TLS#Authentication_support`).

2. Open the `Composer/Context Services` preference page.

3. In the `Security Settings` area, select `Use authentication`.

Enter the username and password and click `Test Connection`.

End of procedure

Upgrading Projects/Diagrams

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

For information, see the Getting Started With Composer book, Upgrading Projects and Diagrams page, on the Composer 8.1 Help Wiki.

Notes:

- 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. See “Routing Upgrade Limitations” on page 68.

Migrating IRD Strategies

Starting with Composer 8.1, you can migrate routing strategies created with Interaction Routing Designer 8.0+ into Composer Projects as SCXML-based workflow diagrams.

For more information, start with the Migration Overview in the IRD to Composer Migration Guide Wiki.

Routing 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 "Configuring Tomcat Settings" on page 52 , you must deploy the Project on Tomcat.

For information, see the Validation, Debugging, and Deployment book, Deploying Composer Applications page, on the `Composer 8.1 Help Wiki`.

Installing the Business Rules Plugin

As described in "Interface with Genesys Rules Engine" on page 31, 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 Rule System 8.1 Deployment Guide*. See Chapter 2, Installation.

- For more information on Composer's business rule components, see the `Busines Rules` book in the *Composer 8.1 Help*. If you install the plugin, you will also have access to the *Genesys Rules System 8.1 Rules Development Tool Help*.
- For additional information, see the *Genesys Rules System 8.1 Rules Authoring Tool Help*.

Chapter

4 Uninstallation

This chapter describes the uninstallation process for Composer. It contains the following section:

Before Uninstalling

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.

Procedure: Exporting projects to a safe location

Start of procedure

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 destination folder to export to.
5. Click the `Finish` button.

End of procedure

Uninstalling Composer

Use the procedure below if you need to uninstall Composer.

Procedure:
Uninstalling Composer

Start of procedure

1. Go to `Control Panel > Add/Remove Programs.`
2. Select `Genesys Composer` from the list of currently installed programs, and then click `Remove.`

   The Genesys Installation Wizard's `Welcome` screen appears.
3. Select `Remove` and click `Next`.
4. Follow the prompts to uninstall Composer.
5. On the `Maintenance Complete` screen, select the option to restart now or later.

End of procedure

Genesys

Supplements

Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

Composer Help Wiki

- The *Composer 8.1 Help,* which is your main source of information for using Composer to develop voice and routing applications, is available on the `Composer Documentation Wiki`. Selecting `Help` > `Help Contents` from within Composer directs new and existing users to the wiki.
- Since Composer is based on Eclipse (`www.eclipse.org`), you should familiarize yourself with basic Eclipse concepts by referring to the *Workbench User Guide* available fromwithin Composer.

Orchestration Server Wiki

- The `Orchestration Server wiki` is the Genesys language specification for the following interfaces: (1) SCXML — What we support from the standard, both from an interface and behavioral standpoint; (2) Domain-specific languages (model modules) for all the Genesys-specific functional modules; (3) External interfaces to platform and SCXML sessions.

Genesys Voice Platform Wiki

- The `Genesys Voice Platform wiki` contains *Genesys VoiceXML 2.1 Reference Help*, which provides information about developing Voice Extensible Markup Language (VoiceXML) applications. It presents VoiceXML concepts, and provides examples that focus on the GVP Next Generation Interpreter (NGI) implementation of VoiceXML.

Routing Applications User's Guide

- This guide assists the first-time Composer user in developing SCXML-based Universal Routing applications. It is intended for both technical and non-technical developers. Its primary goal is to familiarize you with the Composer interface and simplify the process of building routing workflow diagrams.

Cheat Sheets

- Selecting `Help` > `Cheat Sheets` opens a dialog box where you can expand `Composer` and select tutorials to quickly get started with the concepts. This includes tutorials for pre-configurations like for the SIP Phone settings, creating first voice applications, and so on.

Management Framework

- *Framework 8.1 Deployment Guide,* which provides information about configuring, installing, starting, and stopping Framework components.
- *Framework 8.1 Genesys Administrator Help,* which provides information about configuring and provisioning contact center objects by using the Genesys Administrator.
- *Framework 8.1 Configuration Options Reference Manual,* which provides descriptions of the configuration options for Framework components.
- *Framework 8.1 Stat Server User's Guide*, which describes the configuration, installation, and start procedures relevant to deploying Stat Server.

SIP Server

- *Framework 8.1 SIP Server Deployment Guide,* which provides information about configuring and installing SIP Server.

Universal Routing

- *Orchestration Server 8.1 Deployment Guide*. Contains deployment information for Genesys Orchestration Server, which offers an open standards-based platform with an SCXML engine enabling intelligent distribution of interactions throughout the enterprise. Orchestration Server interprets the top-level SCXML document created as a result of an interaction processing diagram created in Composer.

eServices/Multimedia

The *Context Services User's Guide*, available on the Context Services Wiki, which provides information on the Universal Contact Server database of customer-related, service, and interaction-centric data (current and historical). Composer's Context Services blocks use this database.

Genesys Voice Platform

- *Genesys Voice Platform 8.1 Deployment Guide,* which provides information about installing and configuring Genesys Voice Platform (GVP).
- *Genesys Voice Platform 8.1 User's Guide,* which provides information about configuring, provisioning, and monitoring GVP and its components.
- *Genesys Voice Platform 8.1 Legacy Genesys VoiceXML 2.1 Reference Manual*, which describes the VoiceXML 2.1 language as implemented by the Legacy GVP Interpreter (GVPi) in GVP 7.6 and earlier, and which is now supported in the GVP 8.1 release.
- *Genesys Voice Platform 8.1 CCXML Reference Manual,* which provides information about developing Call Control Extensible Markup Language (CCXML) applications for GVP.
- *Genesys Voice Platform 8.1 Troubleshooting Guide,* which provides information about Simple Network Management Protocol (SNMP) Management Information Bases (MIBs) and traps for GVP, as well as troubleshooting methodology.
- *Genesys Voice Platform 8.1 Configuration Options Reference,* which replicates the metadata available in the Genesys provisioning GUI, to provide information about all the GVP configuration options, including descriptions, syntax, valid values, and default values.
- *Genesys Voice Platform 8.1 Metrics Reference,* which provides information about all the GVP metrics (VoiceXML and CCXML application event logs), including descriptions, format, logging level, source component, and metric ID.

Voice Platform Solution

- *Voice Platform Solution 8.1 Integration Guide,* which provides information about integrating GVP, SIP Server, and, if applicable, IVR Server.

Open Standards

- *W3C Voice Extensible Markup Language (VoiceXML) 2.1, W3C Recommendation 19 June 2007,* which is the World Wide Web Consortium (W3C) VoiceXML specification that GVP NGI supports.
- *W3C Voice Extensible Markup Language (VoiceXML) 2.0, W3C Recommendation 16 March 2004,* which is the W3C VoiceXML specification that GVP supports.
- *W3C Speech Synthesis Markup Language (SSML) Version 1.0, Recommendation 7 September 2004,* which is the W3C SSML specification that GVP supports.
- *W3C Voice Browser Call Control: CCXML Version 1.0, W3C Working Draft 29 June 2005,* which is the W3C CCXML specification that GVP supports.
- *W3C Semantic Interpretation for Speech Recognition (SISR) Version 1.0, W3C Recommendation 5 April 2007,* which is the W3C SISR specification that GVP supports.
- *W3C Speech Recognition Grammar Specification (SRGS) Version 1.0, W3C Recommendation 16 March 2004,* which is the W3C SRGS specification that GVP supports.

Genesys

- *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Technical Support for more information.
- Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at `http://genesyslab.com/support`.

Information about supported hardware and third-party software is available on the Genesys Documentation Wiki and on the Genesys Technical website in the following documents:

- *Genesys Supported Operating Environment Reference Guide*
- *Genesys Supported Media Interfaces Reference Manual*

Consult these additional resources as necessary:

- *Genesys Hardware Sizing Guide,* which provides information about Genesys hardware sizing guidelines for the Genesys 7.x and 8.x releases.

- *Genesys Interoperability Guide,* which provides information on the compatibility of Genesys products with various Configuration Layer Environments; Interoperability of Reporting Templates and Solutions; and Gplus Adapters Interoperability.
- *Genesys Licensing Guide,* which introduces you to the concepts, terminology, and procedures relevant to the Genesys licensing system.
- *Genesys Database Sizing Estimator 8.x Worksheets,* which provides a range of expected database sizes for various Genesys products.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

80fr_ref_06-2008_v8.1.001.00

You will need this number when you are talking with Genesys Technical Support about this product.

Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Type Styles

Table 2 describes and illustrates the type conventions that are used in this document.

Table 2: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 79).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where x stands for . . .

Table 2: Type Styles (Continued)

Type Style	Used For	Examples
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index