Composer 8.0

Deployment Guide

About Genesys

Alcatel-Lucent's Genesys solutions feature leading software that manages customer interactions over phone, Web, and mobile devices. The Genesys software suite handles customer conversations across multiple channels and resources—self-service, assisted-service, and proactive outreach—fulfilling customer requests and optimizing customer care goals while efficiently using resources. Genesys software directs more than 100 million customer interactions every day for 4000 companies and government agencies in 80 countries. These companies and agencies leverage their entire organization, from the contact center to the back office, while dynamically engaging their customers. 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

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

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 at the regional numbers provided on page 11. For complete contact information and procedures, refer to the ***Genesys Technical Support Guide***.

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: 80composer_dep_10-2010_v8.0.401.00

Table of Contents

List of Procedures

Preface

Welcome to the *Composer 8.0 Deployment Guide*. This guide describes how to deploy Composer, an Integrated Development Environment (IDE) used to develop applications for:

- Genesys Voice Platform (GVP)
- Genesys Universal Routing Platform

This document is valid only for the 8.0.4 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 75.

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 URS applications, Composer supports editing of SCXML 1.0.

Composer provides real-time debugging capabilities for GVP voice applications. The debugger is integrated with GVP for making test calls, viewing call traces, and debugging applications.

For more information on Composer, see Chapter 2, "Overview" on page 13. For step-by-step instructions on using Composer, see the *Composer 8.0.4 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.

Ideally, you should be familiar with Hypertext Markup Language (HTML), Extensible Markup Language (XML), and VoiceXML concepts in order to use Composer to build GVP applications and State Chart Extensible Markup Language (SCXML) for building routing applications.

Composer provides a wide range of tools to satisfy the needs of a diverse developer population. If you do not wish to write code or use existing code templates, you can build callflows and workflows using Composer's diagram designer where you place, configure, and connect blocks.

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, contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North America and Latin America	+888-369-5555 (toll-free) +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Malaysia	1-800-814-472 (toll-free) +61-7-3368-6868	support@genesyslab.com.au
India	000-800-100-7136 (toll-free) +91-(022)-3918-0537	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp
Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures.		

Document Change History

Changes for Composer 8.0.4

In addition to updating screen captures to reflect Composer 8.0.4, the following information has been added to this document:

- Added section “Builders/Managers” on page 24.
- Under “New Features” on page 25, added features for “Release 8.0.4”.
- Added new operating system support on page 33.
- Updated “Genesys Software Prerequisites” on page 37.
- Added “Routing Applications User’s Guide” on page 76.
- Added Base URL field to “Context Services Preparation” on page 63.
- Updated “Migration” on page 66.

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, for developing:

- Voice applications for Genesys Voice Platform (GVP) 8.1—a software suite that unifies voice and web technologies to provide a complete solution for customer self-service or assisted service.
- Routing applications for Genesys Universal Routing 8.0 platform, which includes:

  Universal Routing Server (URS)—which enables intelligent distribution of voice and multimedia interactions throughout the enterprise.

  Orchestration Server—an open standards-based platform with an SCXML engine, which enables the customer service process. Using dynamic context data and business logic to make decisions about operations, ORS orchestrates customer service across different media channels over time.

Note: 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.

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):

- Pure 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 Universal Routing Server/Orchestration Server 8.0 SCXML Engine/Interpreter:

- Pure 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 workflow in the center editing area.

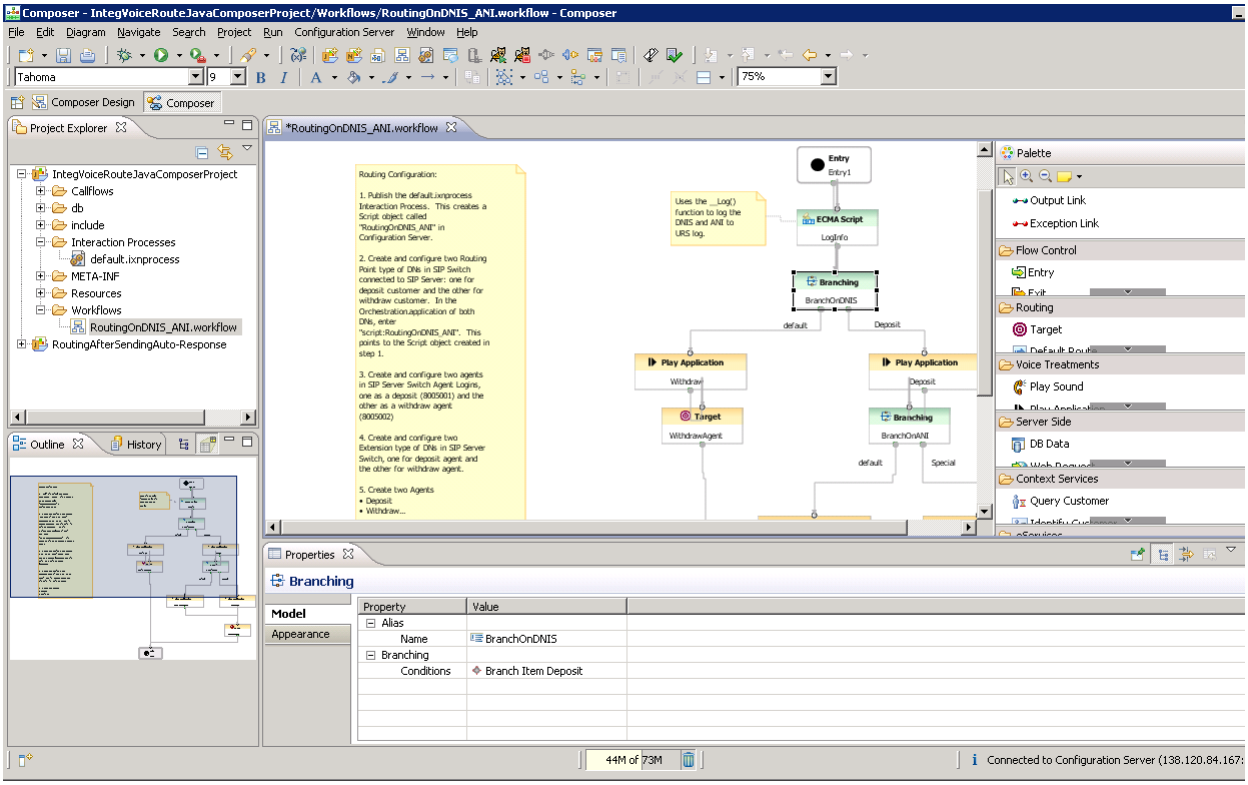

Figure 1: Workflow in Composer

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.

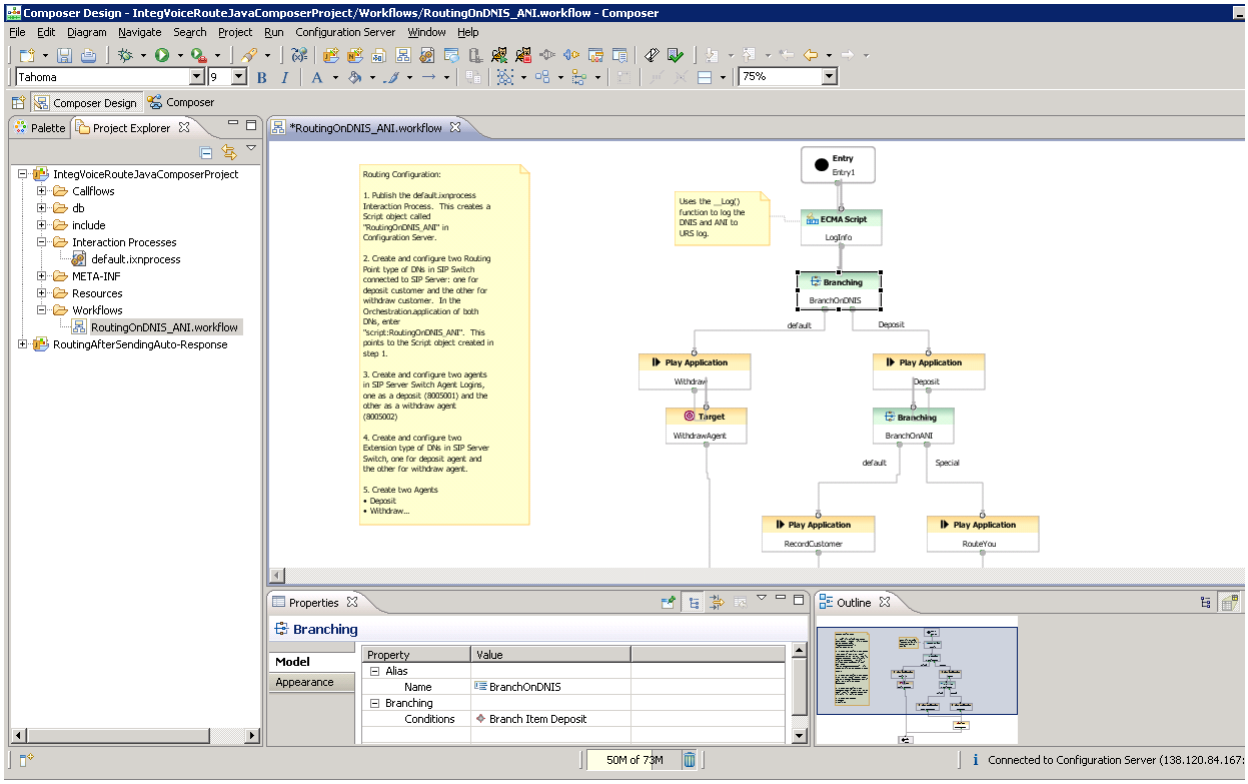

Figure 2: Workflow in Composer Perspective

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.

Perspectives for Voice Applications

Composer includes the following perspectives for building voice applications:

- GVP Debugger perspective, for debugging applications you build or import
- Prompts Manager perspective, which provides the ability to quickly review all prompts in a voice project.

Expression Builder

Composer supplies Expression Builder to easily build expressions that can be used for branching and conditional routing decisions. You can also build ECMAScript expressions that use Genesys supplied SCXML functions documented in the *Universal Routing 8.0 SCXML API Reference* (available from `Help` > `Contents`).

Figure 3 shows an example expression and how to access the Genesys-supplied functions.

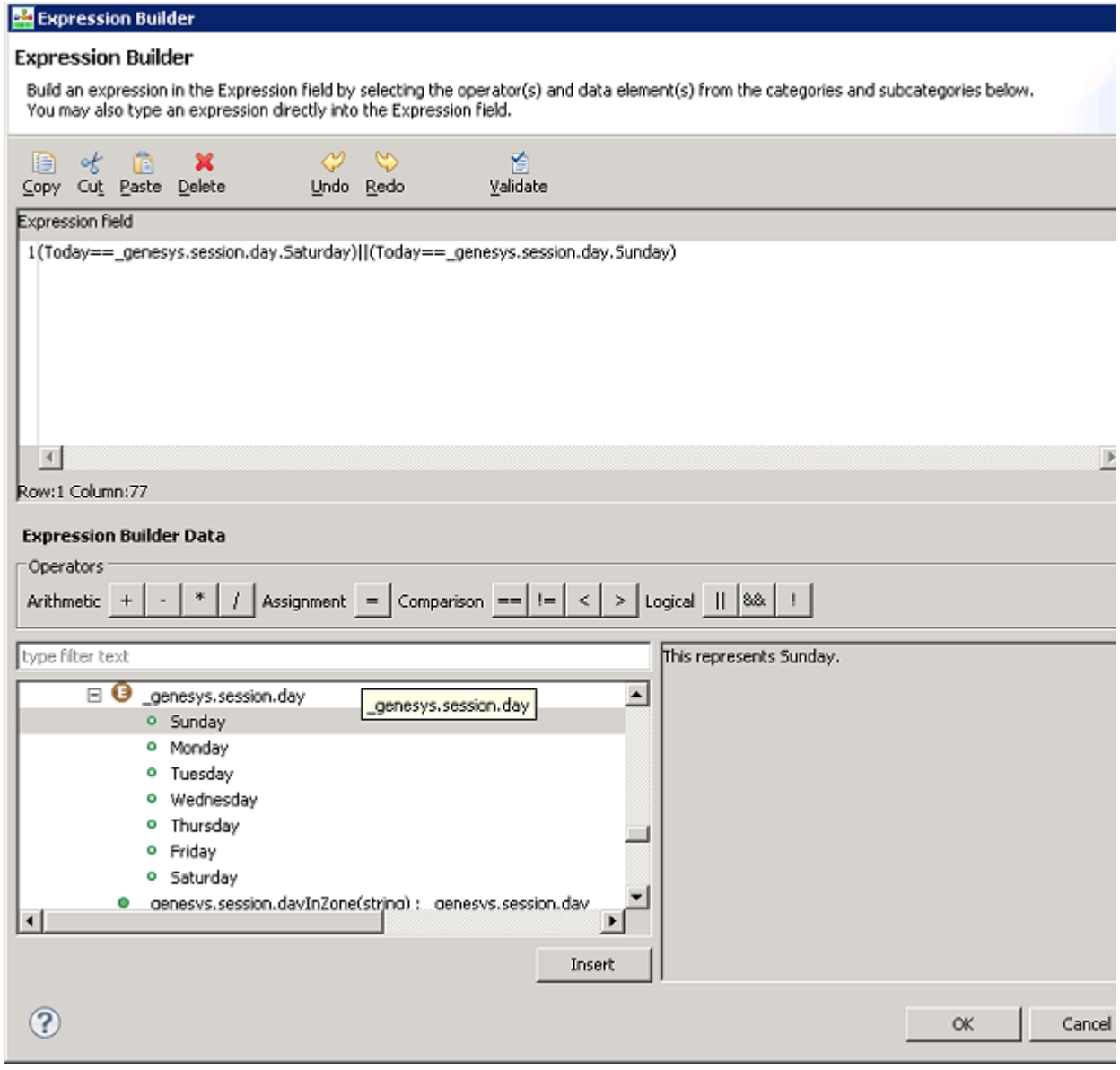

Figure 3: Expression Builder

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.

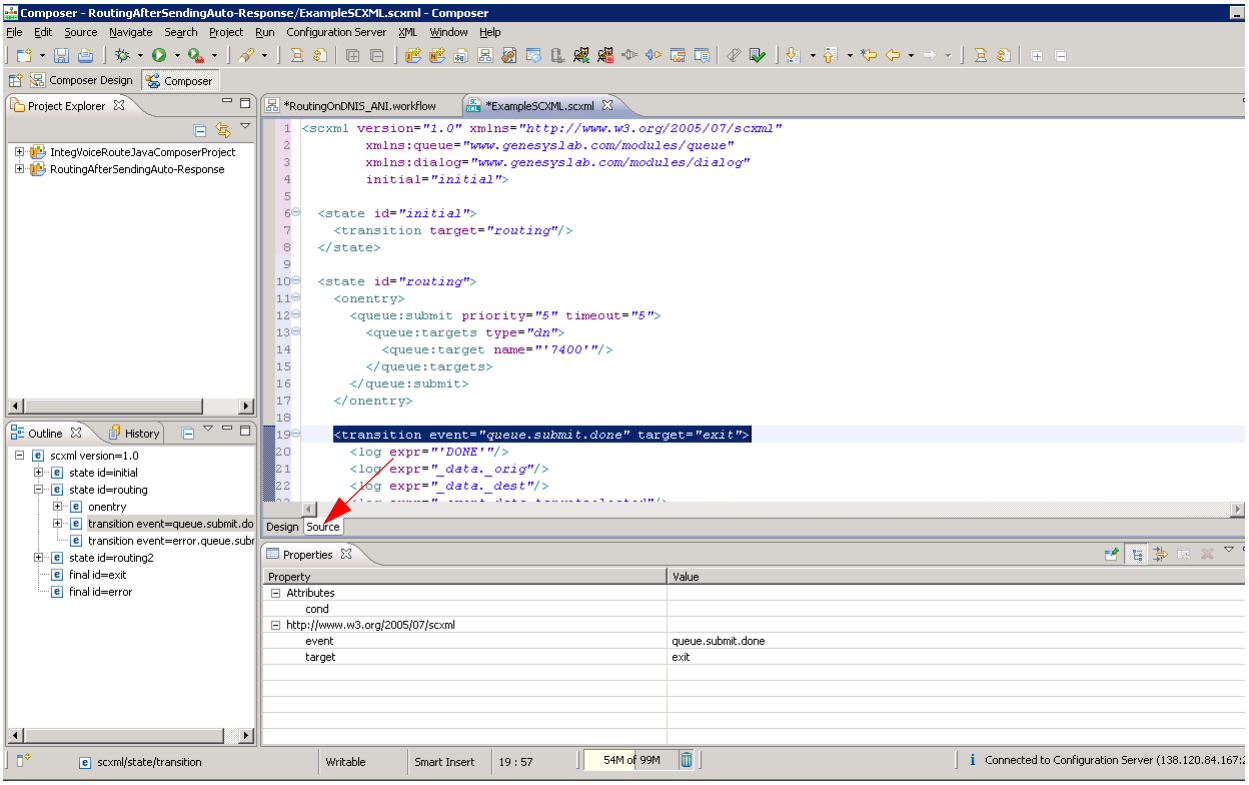

Figure 4: Composer Code Editor, Source View

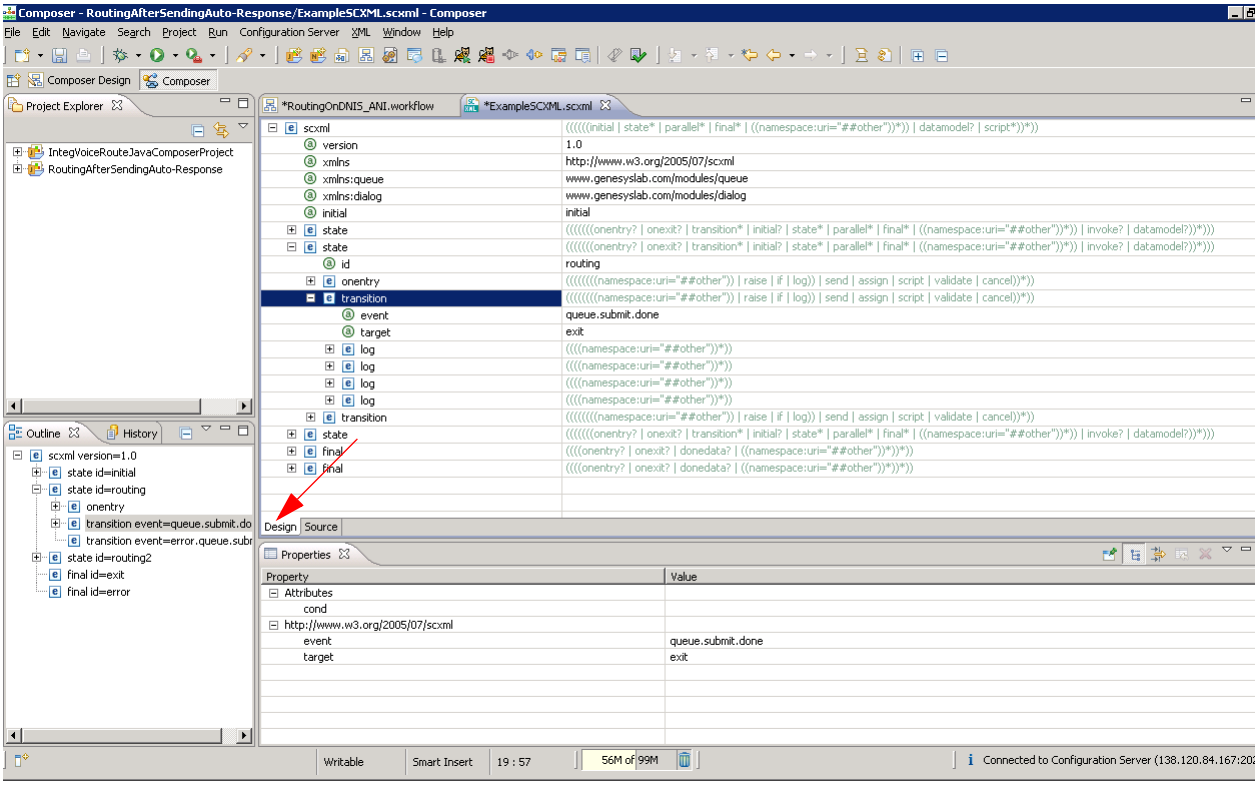

Figure 5: Composer Code Editor, Design View

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.

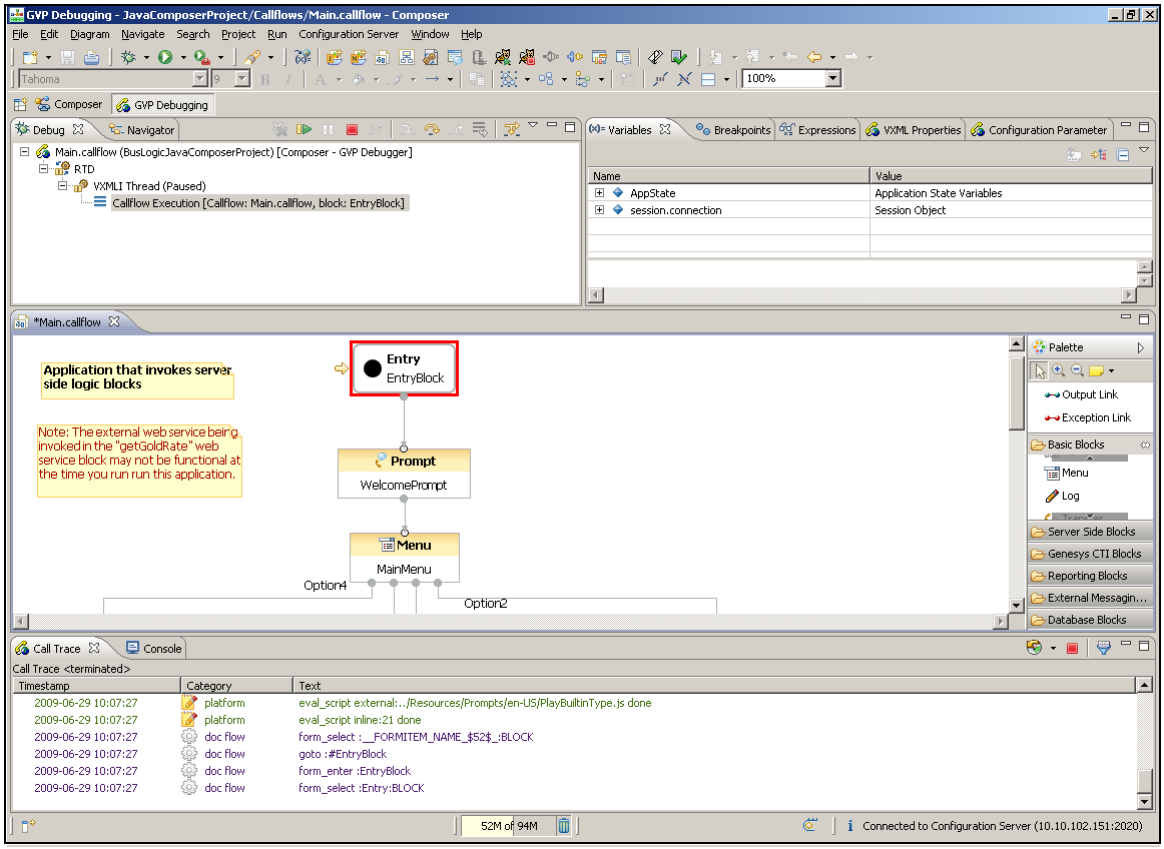

Figure 6: GVP Debugging Perspective

Other Composer Features

Some other Composer main features are summarized below. For details information on all Composer features, see the *Composer 8.0.4 Help*.

Project Templates

Out-of-the-box, reusable Project templates are provided. As shown in Figure 7, 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 URS 8.0 SCXML Engine/Interpreter.

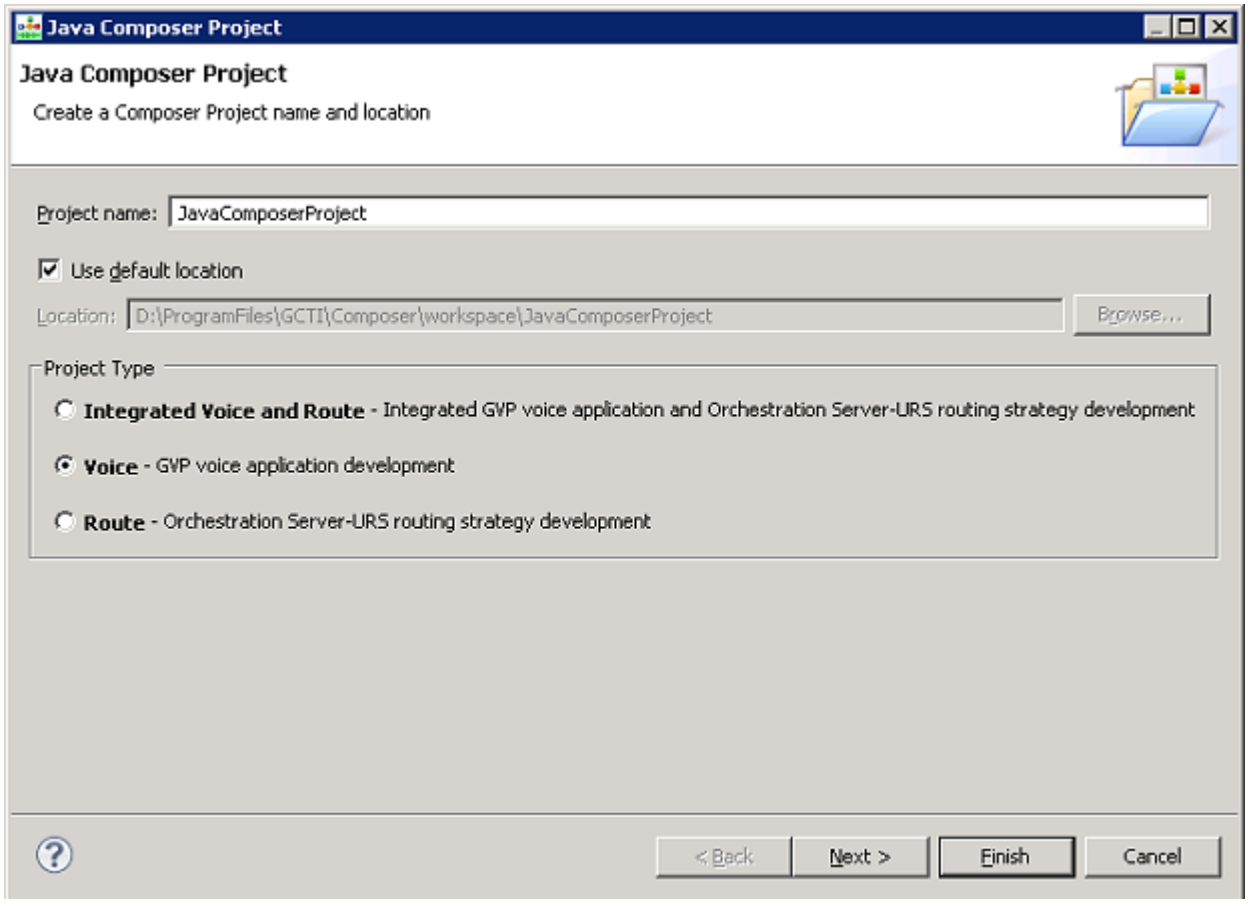

Figure 7: Java Composer Project Wizard

Clicking `Next` brings up available templates for the selected category (see Figure 8).

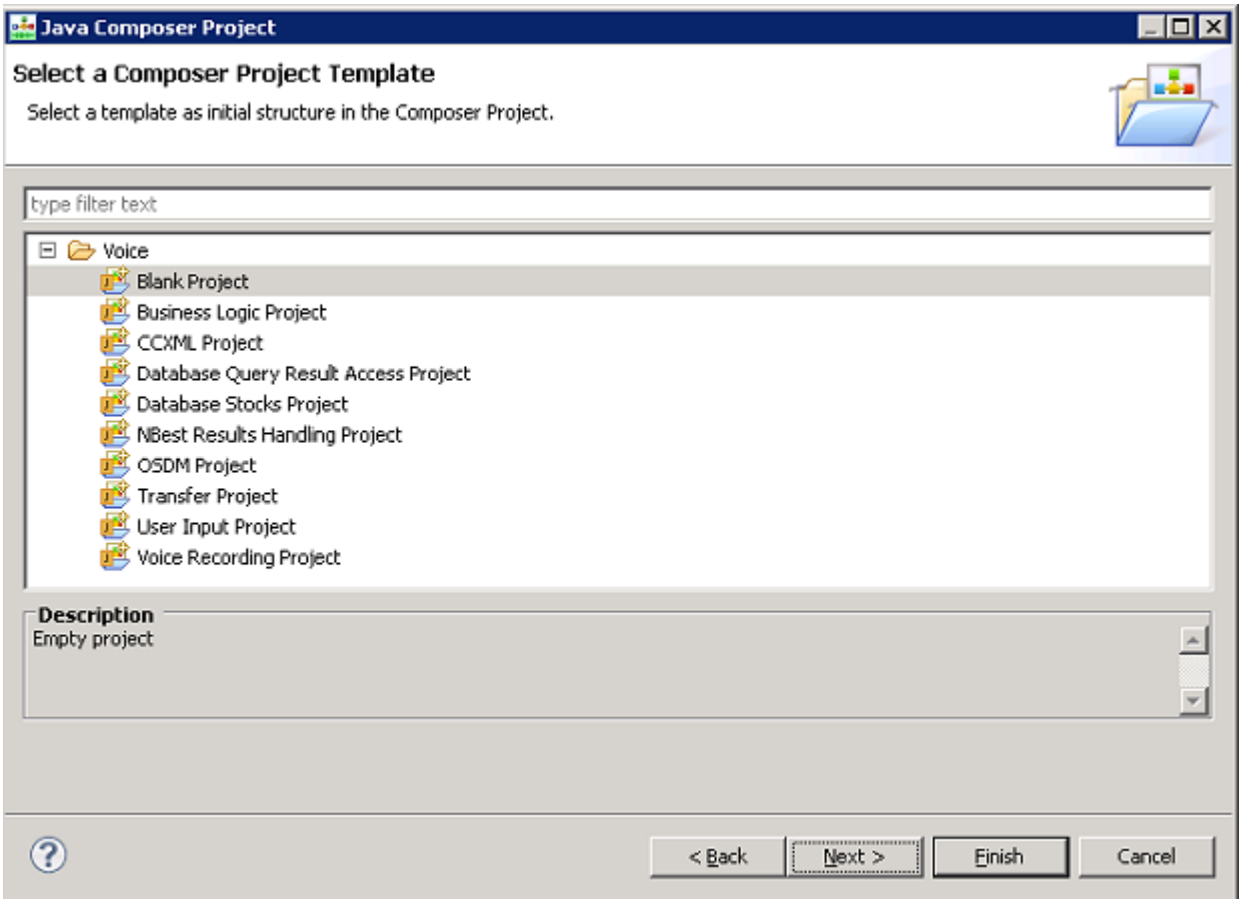

Figure 8: Voice Project Templates

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 voice or routing capabilities" on page 62.

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 8.0 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. 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.

Customization Manager

Customization Manager view helps you manage various aspects of your Composer installation that you have customized. In the 8.0.4 phase, Customization Manager allows you to manage any custom workflow and callflow diagram templates that you have created. You can edit and delete custom templates, add new files, and save diagrams to disk.

New Features

This section describes the new Composer features.

Release 8.0.4

This release continues support for the Context Services (previously "Context Management") capability of Universal Contact Server and its Universal Contact Server Database. If the Context Services capability is enabled at your site, you can now create both SCXML and VXML applications that extract customer data elements from the UCS Database and apply this knowledge as part of a routing or voice self-service application.

Note: Context Services functionality existed in 8.0.3 for SCXML-based routing applications only.

- The following Context Services blocks are now available for VXML applications: Associate Service, Complete Service, Complete State, Complete Task, Enter State, Create Customer, Identify Customer, Query Customer, Query Services, Query States, Query Tasks, Start Service, Start Task, and Update Customer.
- New Context Services blocks for service/offer personalization, reporting, and managing conversations include: Create Customer, Start Task, Complete Task, and Query Task.
- You can map returned data to application variables for the following VXML/SCXML Context Services blocks; Query Customer, Identify Customer, Query Services, Query Tasks, and Query States.

- This release adds support for multi-valued Context Services extension data, which is applicable to all Context Services blocks. Note: Universal Contact Server 8.0.2 or later is required for multi-valued extension support.

Other new features move Composer closer to parity with Universal Routing's strategy creation tool, Interaction Routing Designer (IRD):

- A Routing Rule block gives the ability to use percentage allocation, load balancing, and statistical routing rules for target selection. Available routing rules are those that have already been created in IRD and currently exist in the Configuration Database.
- A Force Routing block allows you to unconditionally route an interaction to the first target type (`ACD Queue`, `Destination Label`, or `Routing Point`) without any other operations.
- The Target block, used for routing voice interactions, adds new properties to support building conditional expressions, such as those used for share agent by service level agreement routing. A new `Threshold` property allows you to build threshold expressions, which can use relational operators, user-defined variables, and the following URS threshold functions:
  - `sdata` for routing conditions based on statistics
  - `acfgdata` for routing conditions based on data stored in Configuration Server Application objects
  - `callage` to return the age of an interaction
  - `lcfgdata` for routing conditions stored in Lists objects

  The Target and Route Interaction blocks add a `Priority` property, which lets you select a variable that contains an expression returning the priority that the interaction will be given in the queue.
- The Branching block now supports segmenting incoming interactions based on call type and/or media type.
- The External Service Block adds a `User Data` property.
- The following blocks add `Wait for Treatment End` and `Request ID` properties: Play Application, Play Message, and User Input.
- The ECMAScript block (through Expression Builder) exposes the following functions:
  - findServiceObjective, which can find/retrieve a Configuration Server service objective for a given combination of Customer Segment, Service Type, and Media Type.
  - priorityTuning, which can adjust the priority of an interaction by taking into account age, expected wait time, and service objective.
- The User Input block adds support for verify digits, retry, success case, and failure case prompts.
- New voice treatment blocks are added:
  - The Pause block inserts a pause between treatments.
  - The Create User Announcement block records an announcement from a user (supports multiple prompts).

  - The Delete User Announcement block deletes an announcement from the Create User Announcement block, possibly in a different workflow.
  - The IVR block sends an interaction to an Interactive Voice Response unit. It has both Compatibility and Non-Compatibility modes to support, for example, specifying a remote resource to be used for a treatment.
  - The Cancel Call block allows you to stop a currently running call.
- The Target block is enhanced to display a busy treatments output port, which can be connected to treatment blocks, such as Play Application, Play Message, Play Sound, and User Input.
- Workbin blocks can now be added to interaction process diagrams. A Workbin block, which represents a temporary storage area for interactions, can be associated with agents, agent groups, places, or place groups.
  - You have the option to define one or more views for a workbin, which defines the conditions for extracting interactions and directing them into workflows. You define these conditions as expressions comprised of logical operators and interaction attributes.
  - These conditions can control the order for submitting interactions, the schedule for submitting, and segmentation (submitting an equal number of interactions of different segments). The conditions can also specify database hints for performance optimization and parameterized conditions for the Supervisor Desktop.
- The Create SMS (Short Message Service) block allows you to specify pre-written text for the content of the SMS.
- The Route Interaction block adds a `Workbin Name` property. Its value will be used as the workbin for targets specified in the block.
- The Interaction Queue, Workflow, and Workbin IPD blocks add an `Object Name` property, which shows the Configuration Server `Script` object name after the object is published.
- A new *Composer 8.0 Routing Applications User's Guide*, available on the Genesys documentation library DVD, introduces new Composer users to the GUI, interaction process diagrams, and workflow diagrams.

Other New Features

- A Customization Manager view helps you manage various aspects of your Composer installation that you have customized. In the 8.0.4 phase, Customization Manager allows you to manage any custom workflow and callflow diagram templates, which you have created. You can edit and delete custom templates, add new files, and save diagrams to disk.
- You can save a callflow or workflow diagram as a template and have the template appear on the list of available templates when creating a new diagram. You can also remove previously added templates. Diagrams saved as templates can exported to/imported from the file system.

- When defining a database connection profile, you can enable connection pooling, which maintains a set of database connections that can be reused for requests to databases. You can use this feature to enhance performance by avoiding time-consuming re-establishment of connections to databases.
- When defining the languages an application supports, you can define custom locales.
- Also, new additional Composer-defined locales are introduced: Hong Kong Cantonese, English-Irish, English-India, English-Scottish, Mexican-Spanish, Icelandic-Iceland, Thai-Thailand, Bengali-India, Spanish-Argentina, United States-Spanish, Gujarati-India, Kannada-India, Malayalam-India, Marathi-India. Oriya-India, Punjabi-India, Tamil-India, and Telugu-India.
- In cases where multiple records are returned, a Looping block, available for both callflows and workflows, can loop through all the records. For each iteration of the loop, mapped variables can be populated with the values of the next record.
- Additional VXML schemas can be added into Composer and used in namespaces for new VXML files created through Composer's VXML editor.
- The Set Call Result block for voice applications allows tagging of calls with SQA call status (success, failure).
- The Disconnect block adds a `Reason` property. The content can be either an ECMAScript expression created in Expression Builder or free-form text.
- Composer adds Microsoft 7 (Premium and Ultimate editions) and Windows Server 2008, 32-bit to its list of supported operating systems.

Release 8.0.3

This release begins support for the Context Services capability of Universal Contact Server and the Universal Contact Server (UCS) Database, a repository of customer-related, service, and interaction-centric data (current and historical). The primary purpose of Context Services is to personalize/orchestrate service delivery to customers and to enhance reporting capabilities.

If Context Services is enabled at your site, you can create SCXML-based applications that extract customer data elements from the UCS Database and apply this knowledge during the routing of interactions or as part of a self-service application. New blocks allow you to create workflow applications that:

- Identify customers and update their profiles
- Extend customer profiles with user-defined information
- Query a customer's profile
- Associate services with customers

- Create/start/complete customer services
- Query customers' active services
- Enter and complete service states
- Query service histories
- Query active and completed service states

A Project template is provided demonstrating the use of Context Services.

This release begins support for SCXML-based routing workflows that process multimedia (non-voice) interactions. New multimedia processing blocks let you define workflows that:

- Create an outbound e-mail
- Perform various types of processing on e-mails including sending
- Route a multimedia interaction to a target
- Place a multimedia interaction in a queue
- Invoke functions through Genesys External Service Protocol
- Create, process, and send a Short Message Service text message
- Send Stop processing information for an interaction to Interaction Server and update the Universal Contact Server Database.

Project templates are provided demonstrating how to create multimedia workflows.

Composer introduces a new type of diagram used for both voice and multimedia, called an interaction process diagram (IPD). When used for multimedia workflows, IPDs can move interactions from media servers (such as E-mail and SMS) to queues, pull interactions from queues, and submit interactions to workflow strategies for specific types of processing. The following new blocks support processing interactions with IPDs:

- Interaction Queue for defining new queues in the Configuration Database.
- Media Server Block for getting interactions from media server endpoints into the IPD.
- Workflow for pointing to a workflow resource, such as a workflow diagram or SCXML file.
- A new type of variable, called a Project variable, allows you to share information across different workflows.

Project templates are provided demonstrating the use of IPDs.

This release also contains some minor enhancements for voice callflow / VXML development:

- Example VXML code template as an example of how to handle N-Best results.
- All Composer blocks that support prompts now support RTSP resources/URIs.
- Variable type prompts can now have an RTSP URI in a variable.

- The Help documents how to create and import prompts record as per the Prompt Manager export specifications.
- A new Entry block property allows suppression of data within the Nuance 9 platform ASR logs.
- Non-numeric strings can be dialed for transfer destinations.
- Support for the Call Progress Analysis/AMD on transfer feature using the Genesys media server, third party media gateways, and the PSTN Connector.
- Support for the AT&T blind transfer with the following options: Out of Band Courtesy, Out of Band Consult, and Out of Band Conference.

Other enhancements that relate to both voice and routing applications include the following:

- You can save a callflow or workflow diagram as an image in one of the following formats: GIF, BMP, JPEG, SVG, PNG, or PDF. You can also export the diagram to HTML.
- You have the ability to hide file types in Composer's `File` > `New` menu.

Release 8.0.2

This release of Composer is compatible with Universal Routing 8.0. It provides the following routing strategy ("workflow") development features, including:

- Authoring of workflows through a drag-and-drop visual designer or direct creation and editing of SCXML files. Universal Routing Server 8.0 is required to execute these workflows. The following categories of blocks for creating routing applications are provided: Flow Control, Routing, Voice Treatment, and Server Side.
- Workflow samples demonstrate segmenting interactions, target selection, percent allocation, statistical routing, web requests, and other functionality.
- Global exception handling, which is available through the Entry block of a workflow.
- An Expression Builder with syntax checking for creating expressions, which can then be used for branching and routing decisions. The Assign, Branching, and ECMAScript blocks access the Expression Builder, which can also be used for voice callflows.
- A Skill Expression Builder with syntax checking for routing based on the value of a skill expression. Routing can also be based on the value of a statistical expression.
- A List Objects Manager to create, for example, lists of 800 numbers that can be accessed by workflows. Key-value pairs can be specified in List elements. List objects are stored in Configuration Server and can be retrieved by a strategy/workflow at runtime.

- A Statistics Manager and Builder, which lets you create custom statistics from the URS predefined statistics. The ability to use the URS predefined statistics in a workflow is also provided.
- A new Composer Design perspective, which provides an interface that facilitates the creation of callflow and workflow diagrams.
- The ability to write ECMAScript within a workflow, to be executed by an application server during runtime. Examples of how ECMAScript can be used include conditional routing, data type, and string manipulation functions. ECMAScript expressions can be created using the Genesys-supplied Functional Modules described in the Composer Help book, *Universal Routing 8.0 SCXML Reference*.
- The ability to connect to Genesys Configuration Server during design time, to access and validate specific Genesys configuration objects that are referenced in workflows. Composer also supports working in an offline mode, when it is not connected to Configuration Server.

This release of Composer is compatible with Genesys Voice Platform (GVP) 8.1 release features. It includes the following new features for voice applications:

- Web services stubbing, which allows you to work with Web Services in an "offline" mode when you do not have access to the Web Service itself or if the Web Service is under development (used for both routing and voice applications).
- A new Query Builder for use with the DB Data block, in both voice application callflows as well as routing strategy workflows.
- You can now visually work with and execute database stored procedures in the DB Data block.
- You can select the default and active locales to use for a project when creating a new project through Composer's Project wizard. This will drive the available locales within Composer's Grammar Builder and Prompts Manager. Additional locales may be added to the project at any time.
- Import and export of Composer projects as well as individual routing strategy files is supported.
- Full support for migrating/upgrading 8.0.1 Composer Voice Projects and callflow diagrams.
- The ability to create and then import and export custom blocks.
- You can now start and stop the bundled Tomcat server from within Composer.
- Support for catching `error.com.genesyslab.subdialog.maxdepthexceeded`.
- Support for catching Call Progress Analysis events through custom event handling.
- Support for offboard Dual Tone Multi-Frequency (DTMF) signal recognition.

- A new Prompts Manager perspective makes it easier to review prompts for any application.
- A new preference for prompt validation indicates if a validation check for missing prompts must be enabled.

The following new features apply to both voice applications and routing strategy workflows:

- The ability to hide voice application and/or routing workflow development capabilities through a Composer preference setting.
- Keyboard navigation in all dialogs. User entry forms and keyboard mnemonics/hot keys are provided for common tasks.
- Block look and feel has changed to make it easier to visually identify blocks.
- Support for defining custom events.
- HTTPS support is provided for the Web Services and Web Request blocks.
- Genesys supports running Composer as a virtual image using the VMware software and player.

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 1 or later
- Microsoft Windows XP with Service Pack 2 or later
- Microsoft Windows Vista
- Microsoft Windows 7
- Microsoft Windows Server 2008, 32-bit

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 Universal Routing/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 33).

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 35.

Note: For developing SCXML applications and related resources that will be executed on the Genesys Universal Routing Server/Orchestration Server platform, only specific Web application servers are supported. Please refer to the *Orchestration Server 8.0 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 48
- "Configuring proxy settings in Tomcat 6.0" on page 56

Internet Information Server (IIS)

See the following sections ahead:

- "Configuring in IIS Manager" on page 49
- "Configuring IIS Preferences" on page 51
- "Adding MIME Types" on page 52
- "Proxy Configurations for .NET Composer Projects" on page 57

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 8 on page 23) 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.

Databases Supported

Composer 8.0 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.

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 6.0 Service Pack 1 or later
- Microsoft Internet Explorer 7.0
- Mozilla Firefox 2.0, 3.0

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)

Genesys Software Prerequisites

To obtain the full functionality of Composer 8.0.4, the following Genesys products/software components are required:

- Universal Routing Server (URS) 8.0.1 and Orchestration Server (ORS) 8.0.1. SCXML applications created by Composer 8.0.3 and later require ORS 8.0 and URS 8.0 to execute.

- Genesys Voice Platform (GVP) 8.1.3 and Media Control Platform 8.1 for testing VXML applications.
- If you wish to process multimedia interactions, you will need eServices (formerly Multimedia) Interaction Server 8.0.1 You will also need the servers applicable to the media types being processed, such as Genesys E-mail Server 8.0 (formerly E-mail Server Java).
- If you wish to use the Context Services capability of Universal Contact Server in routing workflows and voice callflows, you will need Universal Contact Server 8.0.2. Multi-valued customer profile extensions are supported starting with UCS 8.0.2.

Note: As described in the *eServices (Multimedia) 8.0 Deployment Guide*, the UCS 8.0.2 setup requires that you first install the MCR Java Environment and Libraries for eServices.

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 8.0 Release Advisory.

Note: Genesys does not recommend installation of its components through a Microsoft Remote Desktop connection. You should perform the installation locally.

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 33 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 `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.

3. 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.

4. 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 "Migration" on page 66.

The `Composer Parameters` screen appears.

5. In the `Port` text box, enter the port that will be used to handle Tomcat (for example: `9002`). Click `Next`.

   The `Ready to Install` screen appears.

6. Click `Install`. The `Installation Status` screen appears.

7. 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 8.0 > 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

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 8.0.4 Applications` or

   `C:\Users\<your user>\MyWorkspaceFolder`

End of procedure

Viewing a Sample Application

Composer provides a set of predefined Project templates (see Figure 8 on page 23) 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:
Using a Project template for a 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 9).

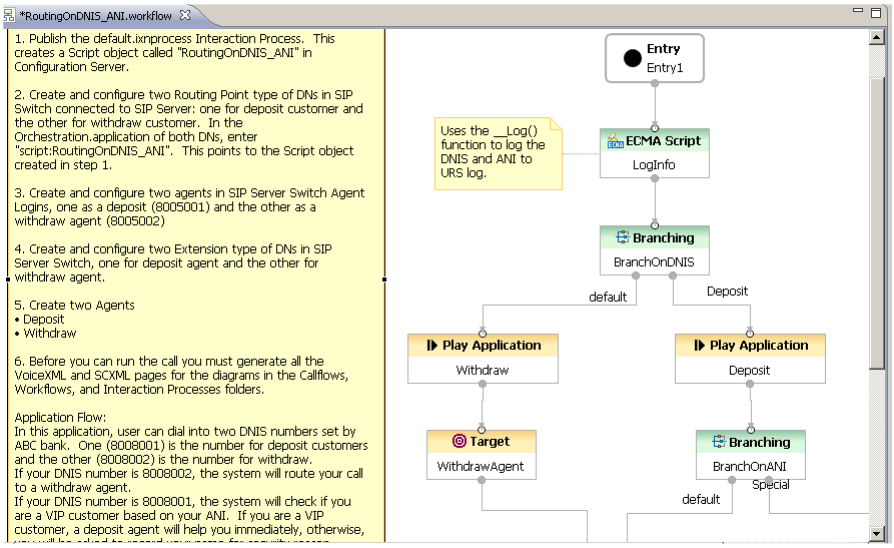

Figure 9: Sample Application Integrated Voice and Route Application

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 (described under “Release 8.0.3” on page 28).

Procedure: Using a Project template for a 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 10).

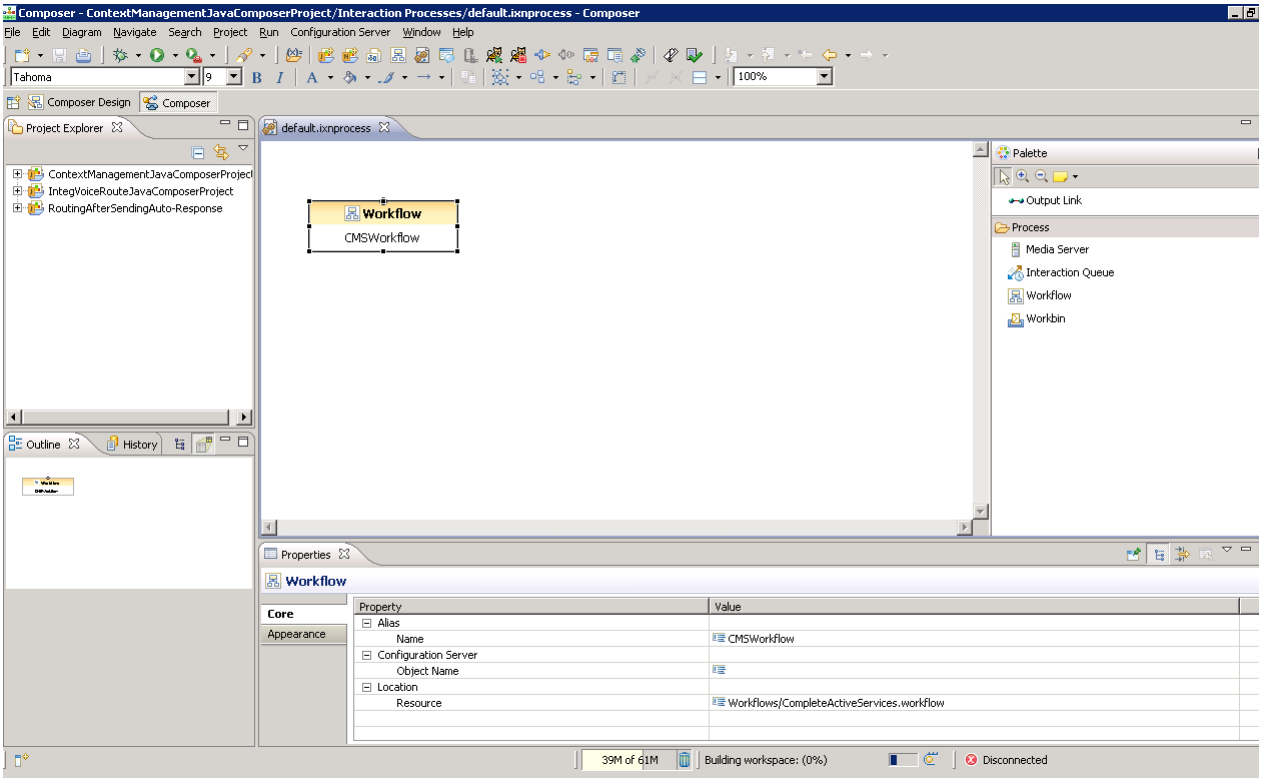

Figure 10: Sample Routing Application

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. An example entry is shown in Figure 11.

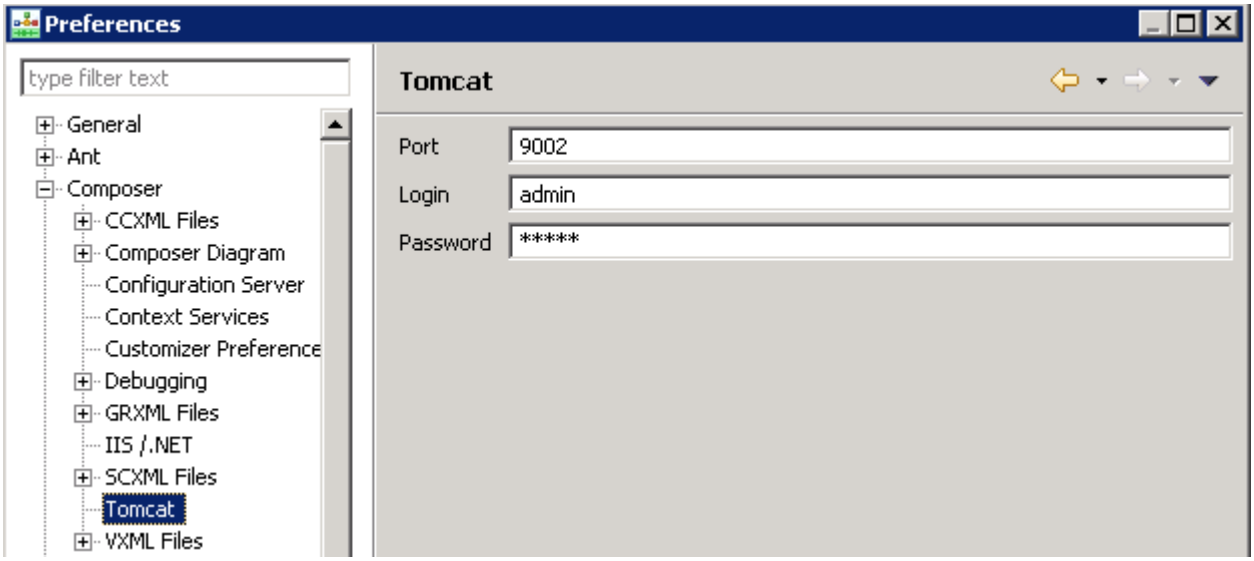

Figure 11: Tomcat Preferences

Procedure: Configuring Tomcat settings

Start of procedure

1. From the `Window` menu, select `Preferences` if the `Preferences` window is not already open.
2. In the left panel, expand the `Composer` node and select `Tomcat`.
3. In the `Port` field, specify the same Tomcat port as you used during product installation.
4. In the `Login` field, enter the default login: `admin`.
5. In the `Password` field, enter the default password: `admin`.
6. Click `Apply`.

End of procedure

Configuring in IIS Manager

Before you can start to create a .NET Composer Project that you will deploy later on a Microsoft IIS application server, you must do the following:

- Configure IIS settings for Composer (see page 51).
- Allow the ASP.NET Web Service extension in Internet Information Services (IIS) Manager (see Figure 12).

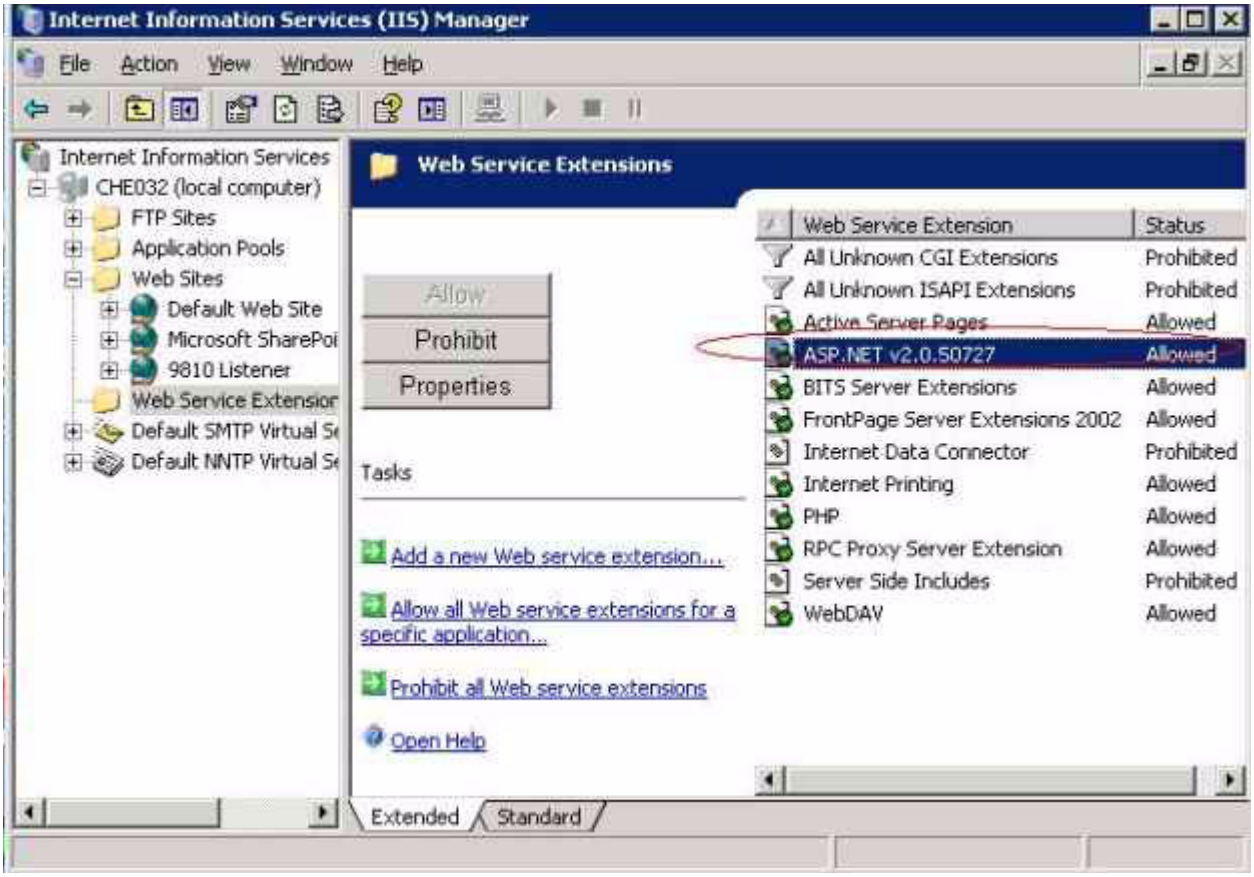

Figure 12: Allow ASP.NET Web Service Extensions in IIS Manager

If running on Vista, during the installation of IIS, IIS Metabase and IIS 6 configuration compatibility must be installed (see Figure 13).

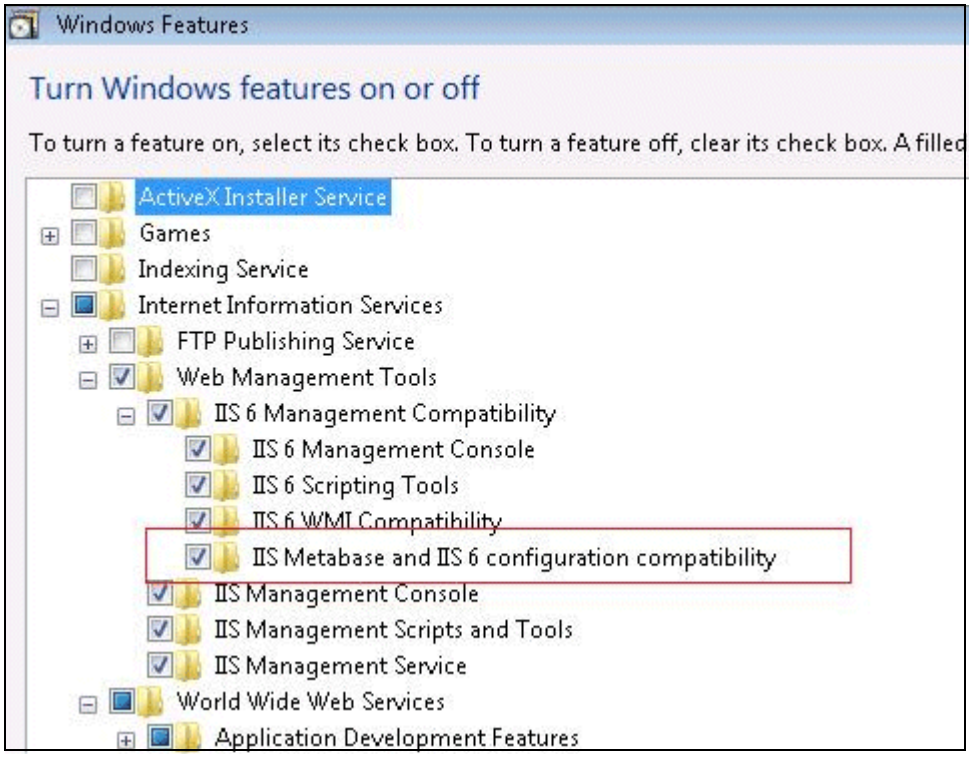

Figure 13: IIS Metabase and IIS 6 Configuration Compatibility

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 12.
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

Next Steps

- Configure IIS settings for Composer.

Configuring IIS Preferences

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

Procedure: Configuring IIS Preferences.

Start of procedure

1. On the main Composer menu, select `Window` > `Preferences`.
2. In the left panel, expand the `Composer` node and select IIS/.NET.An example completed screen is shown in Figure 14.

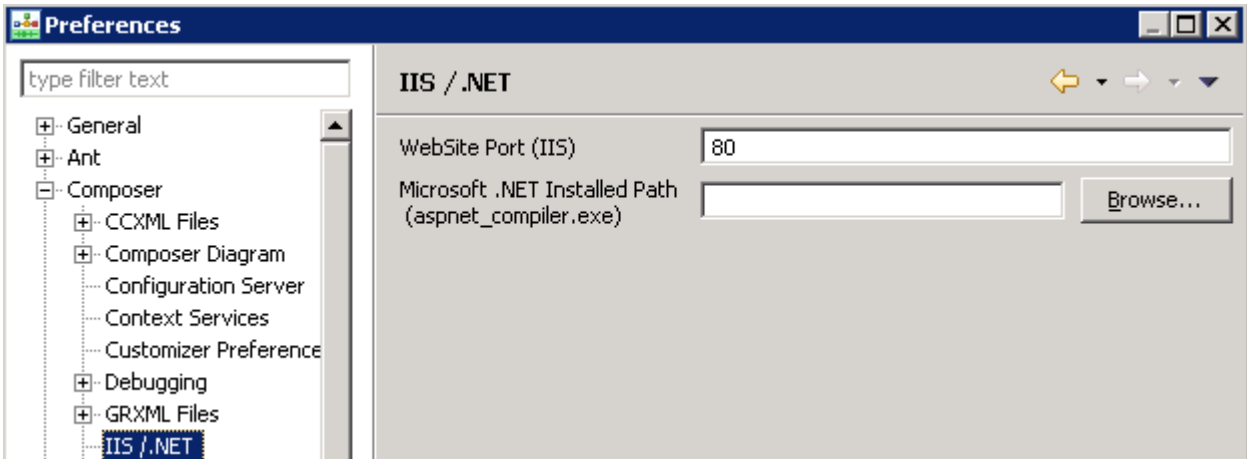

Figure 14: IIS Preferences

3. In the `Website Port (IIS)` field, specify the port number of the website on IIS on which you want to deploy your voice projects. Typically, this value would be `80`, which is the port for the `Default Web Site` in IIS.
4. In the `Microsoft.NET Installed Path (aspnet_compiler.exe)` field, enter the location in which you installed the .NET Framework that contains the ASP.NET compiler. Typically, this value is the following:

   `C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_compiler.exe`
5. Click `Apply`.

End of procedure

Next Steps

- Add Multipurpose Internet Mail Extensions (MIME) types in IIS.

Adding MIME Types

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 perform the debugger configuration. 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.

Procedure: Configuring the GVP Debugger

Start of procedure

1. On the main Composer menu, select `Window` > `Preferences`.
2. In the left panel, expand the `Composer` > `Debugging` and select `GVP Debugger`.

An example is shown in Figure 15.

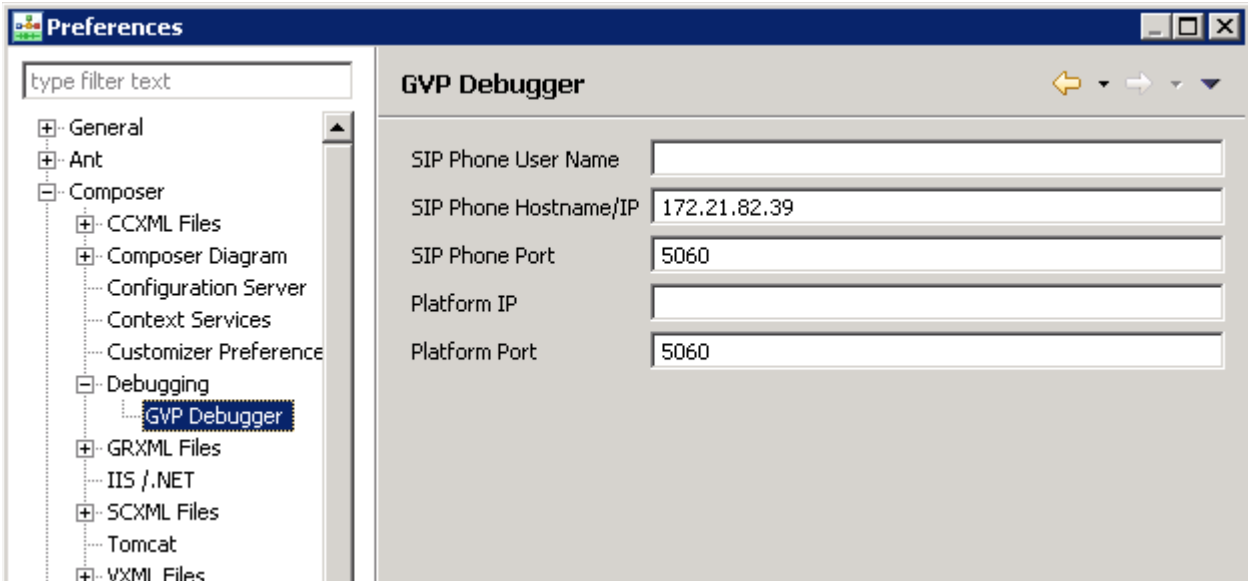

Figure 15: GVP Debugger Preferences

You will need a SIP phone to make test calls.

3. Specify the default values for:
   - `SIP Phone User Name`
   - `SIP Phone Hostname/IP`. This is the address of the computer on which the SIP phone is running
   - `SIP Phone Port`. Typically, SIP phones run on port `5060` or `5070`. Check the settings on your SIP phone, and provide the correct information.
4. Set the `Platform IP` address and `Platform Port` for the GVP Server.

   Typically, the `Platform Port` will be the default port `5060`, or the port that you configured for the Resource Manager (RM), or the default port `5070`, or the port that you configured for the Media Control Platform (MCP) on your GVP Server. You can make *direct calls* to MCP from the Debugger.
5. Click `Apply` and then click `Yes` in the `Propagate Changes` dialog box.

End of procedure

Enabling Debugging in the Media Control Platform (MCP)

To use the debugging feature of Composer, the Media Control Platform (MCP) must be configured to enable debugging.

Procedure: Enabling debugging in the MCP

Start of procedure

1. Open Genesys Administrator for the Configuration environment that is serving the MCP platform.
2. Under the `Provisioning` tab, select `Environment` > `Applications` and open the `Application` object that corresponds to the MCP.
3. Select the `Options` tab and open the section that is called `vxmli`.
4. Change the value of `debug.enabled` to `true`.
5. Restart the MCP application.
6. Log out from Genesys Administrator.

End of procedure

Configuring TCP Ports

Procedure: Ensuring TCP ports have been opened

Purpose: To develop and test Composer applications.

Start of procedure

1. If you have a local firewall on the development server (for example, Windows Firewall on Windows XP/Windows Server 2003), make sure that the following TCP ports have been opened:
   - **Tomcat port** (generally, this is set to port `8080`). If you installed Tomcat on a different port, open its corresponding port in the firewall.
   - **IIS port** (generally, this is set to port `80`). If you installed IIS on a different port, open its corresponding port in the firewall.
   - The **UDP port** on which your SIP phone is running (by default, this will be either `5060` or `5070`). Check your SIP phone settings for the exact port number.
   - **RTP ports** on which your SIP phone will get the audio stream. Check your SIP phone Help file for details on this. Some SIP phones will autoconfigure this during installation.

End of procedure

Next Steps

- If you continue to run into problems with the firewall and calls are not successful, try turning off the firewall temporarily when you make the test calls.

Configuring Proxy Settings

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

Local Proxy

Procedure: 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 16.

Figure 16: Example Proxy Settings in web.config

End of procedure

Prompt Resource Validation

This preference enables diagram validation warnings where prompt audio resources no longer exist in the given file path. If the audio file is no longer present, the diagram block will show a warning icon.

Procedure: Enable prompt resource validation

Start of procedure

1. On the main Composer menu, select `Preferences` if the `Preferences` window is not already open.
2. Select `Composer` > `Composer Diagram`.
3. Select the option `Enable Validation for Prompt Resources`. By default, the preference is not enabled.

End of procedure

Predefined Statistics Creation

A preference controls whether or not to create URS (Router) predefined statistics when connecting to the Configuration Server. If you will be creating strategies that route to targets based on the value of a statistic, you will want to set this preference.

Procedure:
Creating URS predefined statistics

Start of procedure

1. On the main Composer menu, select `Window` > `Preferences`
2. Expand `Composer` > `Configuration Server`.
3. Select the option to `Create router predefined statistics when connecting to the Configuration Server`.
4. Click `Apply` and `OK`.

End of procedure

Connecting to Configuration Server

You may develop URS applications:

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

Whether to connect depends on what you wish to do. For example, you would need to connect to Configuration Server in order to access actual Configuration Database objects through the Target block. Otherwise, if you are working in an offline mode, you can manually type the names of configuration objects in the Target block. Once you connect to Configuration Server, Composer can then validate that these configuration objects actually exist in your Configuration Database, and warn if there are mismatches.

Procedure:
Connecting to Configuration Server

Start of procedure

1. From the Composer main menu, select `Configuration Server` > `Connect`.
2. Enter the user name, password, application name, host, and port information for the Configuration Server used in your environment.
3. Click `Next`.
4. Select the tenant. For a single-tenant environment, select `Environment`.
5. Click `Finish`. Composer can now access Configuration Server data during validation (if configured to do so) and other operations.

End of procedure

ORS and Routing Point Configuration

In addition to specifying the HTTP request parameters, both Universal Routing Server (URS) and Orchestration Server (ORS) must be properly configured. In addition to specifying HTTP request parameters as described below, the URS configuration option `strategy` must be set to `ORS`. This ensures that URS is prepared to process interactions according to requests received from ORS.

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

- *Universal Routing 8.0 Deployment Guide*, Orchestration Support chapter.
- *Universal Routing 8.0 Orchestration Server Deployment Guide*, SCXML Strategy Support and Configuring Orchestration Server chapters.

Procedure: Specifying the HTTP request parameters

Purpose: To inform Orchestration Server of the Application Server URL and other parameters.

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 the http address of the SCXML strategy. You may need to wait until the SCXML strategy is deployed to an Application Server in order complete this step.

   For complete details on this step and other routing configuration details, consult the *Orchestration Server 8.0 Deployment Guide*, SCXML Strategy Support chapter.

   Important! If you have both Composer and Interaction Routing Designer set up in the same environment, check in IRD's `Loading View` that you have not loaded an IRD 7.x 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 Play Sound Block

Perform these steps if using Stream Manager to play treatments via the Treatment blocks for strategies (such as Play Sound). For more information on Stream Manager, start with the *Framework 7.6 Stream Manager Deployment Guide*.

Procedure: Configuration for Play Sound Block

Start of procedure

After installing Stream Manager:

1. Set up a SIP Switching Office and a SIP Switch.
2. Set up a SIP T-Server with an association to the SIP Switch.
3. For your SIP T-Server, ensure that the `sip-port` option under the `TServer` section is unique in your environment.
4. Make sure there is a connection between your SIP T-Server and Stream Manager.
5. For Stream Manager options, in the `contact` section, make sure the SIP port is unique in your environment.
6. On your SIP Switch, create a DN (usually named `Stream Manager`) of type `Voice over IP Service` to enable Stream Manager to properly play the treatments. For information on Stream Manager and the Voice over IP Server type DN, refer to the *Voice Platform Solution 8.1 Integration Guide*.
7. In the `Annex` tab of this DN, add a section called `TServer` with the following options:
   - Name: `contact`, Value: <*`IP address of Stream Manager`*>:<*`SIP Port of Stream Manager`*>
   - Name: `service-type`, Value: `treatment`

   If using a SIP softphone, you will need DN of type `Trunk`. In the `Annex` tab, add a section called `TServer`.
   - Name: `contact`
   - Value: <*`IP address of where SIP softphone is running`*>

End of procedure

Hiding Capabilities

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

Procedure: Hiding voice or routing capabilities

Start of procedure

1. On the main Composer menu, select `Window` > `Preferences`.
2. Expand `General` and select `Capabilities`.
3. Click the `Advanced` button.
4. In the `Advanced Capabilities` dialog box, expand `Composer` (see Figure 17).

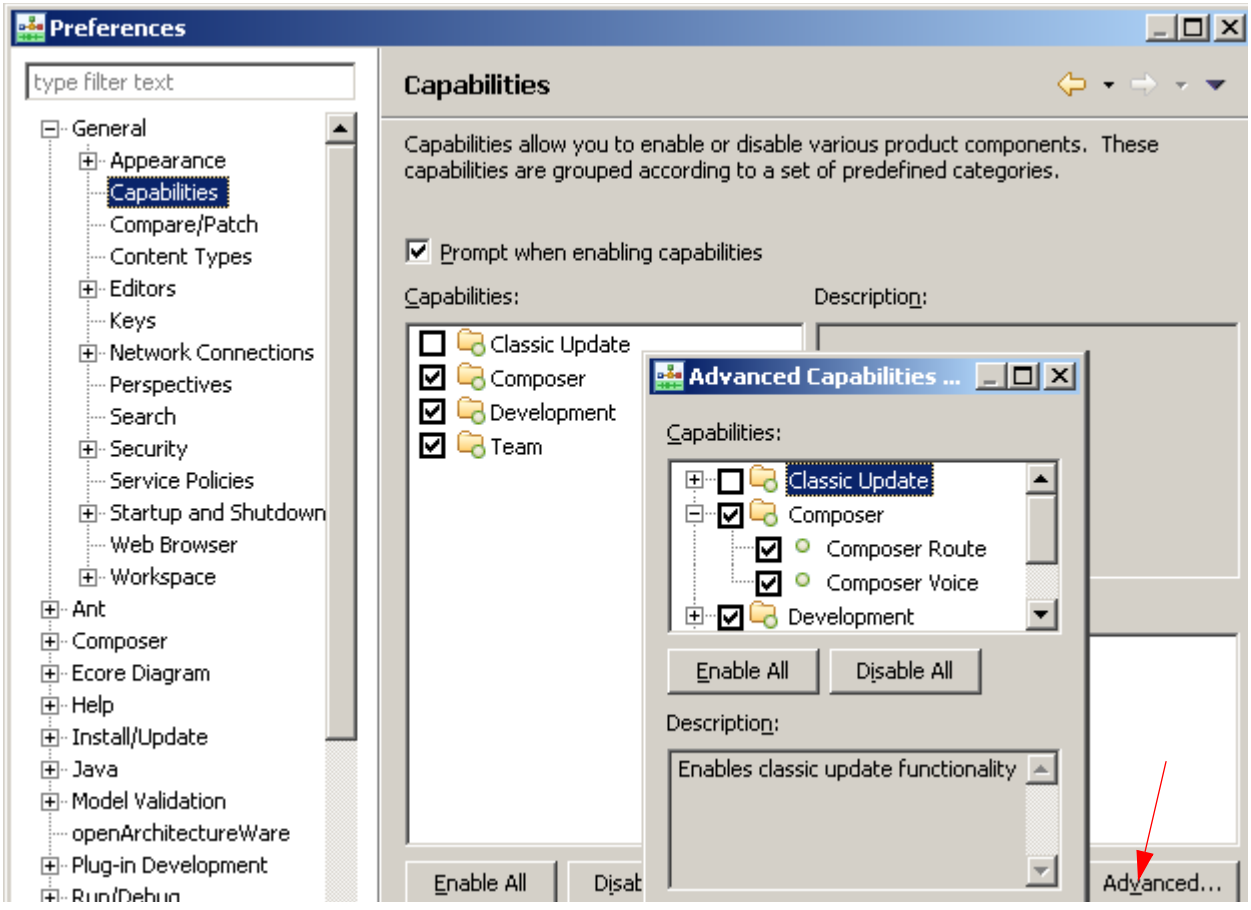

Figure 17: Capabilities Preferences

5. Check or uncheck `Composer Route` or `Composer Voice` based on your need.

- If you uncheck `Composer Voice`, the ability to create projects and diagrams with callflows will no longer be available. Also, perspectives and views exclusive to callflows will not be available. This means you temporarily won't be able to design voice applications for GVP until you re-enable `Composer Voice` capability.
- If you uncheck `Composer Route`, the ability to create projects and diagrams with workflows will no longer be available. Also, perspectives and views exclusive to workflows will not be available. This means you temporarily won't be able to design routing applications for Universal Routing 8.0 until you re-enable `Composer Route` capability.

6. Click `OK` in both dialog boxes.

End of procedure

Context Services Preparation

If you plan to use Context Services (see page 28), follow the procedure below before working with Composer's Context Services blocks.

Procedure: Preparing to use the Context Services blocks

Start of procedure

1. Install Composer 8.0.4.
2. Go to `Window` > `Preferences` > `Composer` > `Context Services`.

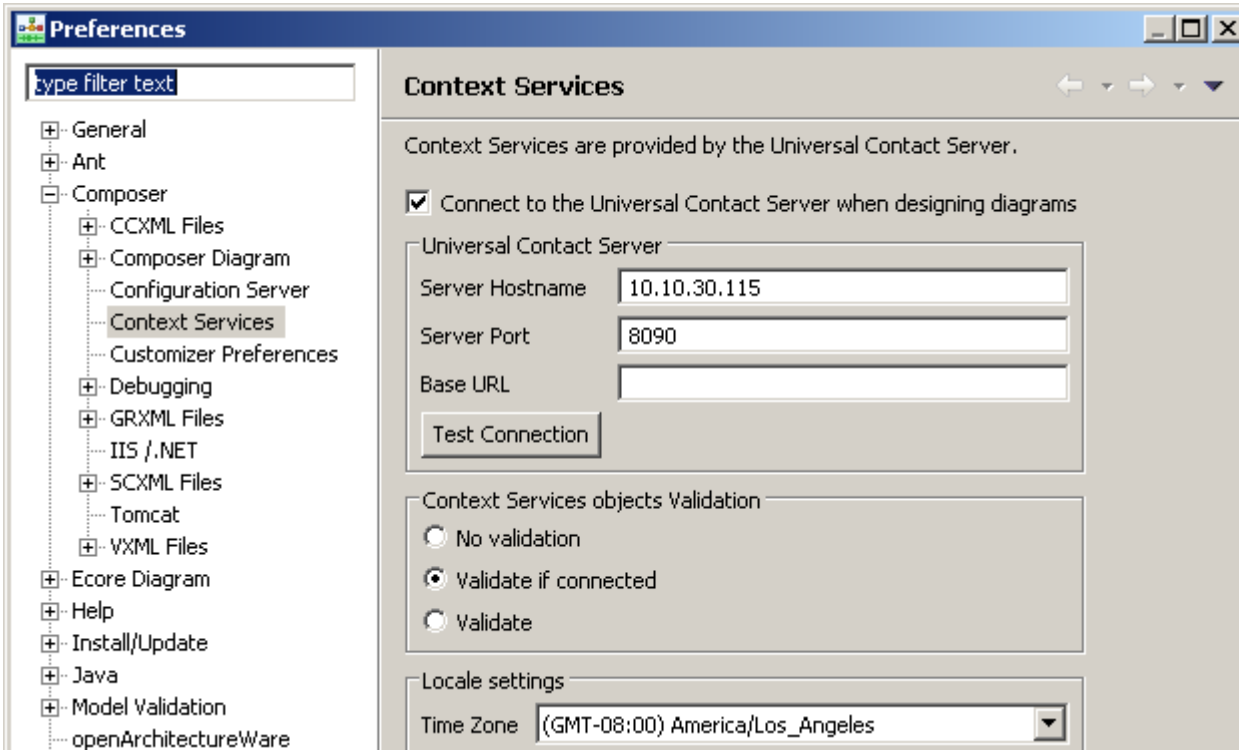

Figure 18: Context Services Preferences

3. Check the following box: `Connect to Universal Contact server when designing diagrams`.
4. Under `Universal Contact Server`, enter the host name.
5. Enter the port number for Universal Contact Server.
6. Enter the `Base URL` for the Context Services server (UCS). The GVP Debugger provides this URL to the VXML platform at design time.
7. Click the `Test Connection` button. Clicking should cause `Connection OK` to appear. If not, check that Universal Contact Server is running and that the entered host/port values are correct.
8. Under `Context Services objects Validation`, select a validation preference.
9. Under `Locale settings`, select your time zone.
10. Click `OK`.

End of procedure

Using Context Services Host and Port Information

Follow the procedure below to verify that Context Services data is being correctly retrieved.

Procedure:
Using Host and Port Information for Context Services

Start of procedure

1. Create a new Java Composer Project (`File` > `New` > `Java Composer Project`), the type used for routing workflows.
2. Create a new workflow diagram as described in the help (`Help` > `Help Contents` > `Composer`).
3. Add an Entry block, an Identify Customer block, and an Exit block. Connect them.
4. In the Entry block, create two variables: `CVHost` and `CVPort`. Set their values to the UCS host and port values previously entered that should be used at runtime to connect to Universal Contact Server.
5. Set the Identify Customer block properties, and generate code.
6. Deploy the SCXML application as described in the *Universal Routing 8.0 Deployment Guide* and *Orchestration Server 8.0 Deployment Guide.*
7. Make a call. Use the log to check that data was correctly fetched. Or you can add an additional ECMAScript block to manipulate the data returned by the Identify Customer block.

End of procedure

Migration

While working with Composer 8.0.4, if you want to use a previously-created Composer Project and Project diagrams, an upgrade/migration is required.

If you simply copy diagrams into a new Composer Project instead of migrating the Project itself, then you must use the diagram upgrade procedure as described in the sections below.

Notes:
- Genesys recommends that you create a dedicated workspace for 8.0.4 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.
- Previously created workflow diagrams cannot be migrated. See "Routing Migration Limitations" on page 70.

Migration Summary

A summary of the Composer migration process is as follows:

1. Obtain Composer 8.0.4 through Genesys Technical Support.
2. Uninstall the older version of Composer. Before un-installing the older version of Composer:
   a. Make a copy of your Composer workspace folder (which contains all your Projects and Project files), as your workspace may be deleted if it is located under the installation directory (`C:\Program Files\GCTI\Composer 8.0\workspace`).
   b. Uninstall the older version of Composer.
3. Install Composer 8.0.4.
4. Migrate at a Project level or at the Diagram level as described below.

Procedure: Project Level Migration

Project level migration will automatically apply Diagram level migration for all the diagram files directly residing within the diagram (`Callflows`) folder.

An `archive` folder will be created in the Project with a backup archived copy of the original Project content.

Start of procedure

To migrate a Project created in a previous version:

1. Import an old Composer Project into Composer's Project Explorer view. From the menu, select `File` > `Import`.
2. In the Import dialog, navigate to `General` and double-click `Existing Projects into Workspace`.
3. Browse to the Composer Project location and select the Project(s).
4. Mark the checkbox `Copy projects into workspace` (see Figure 19).

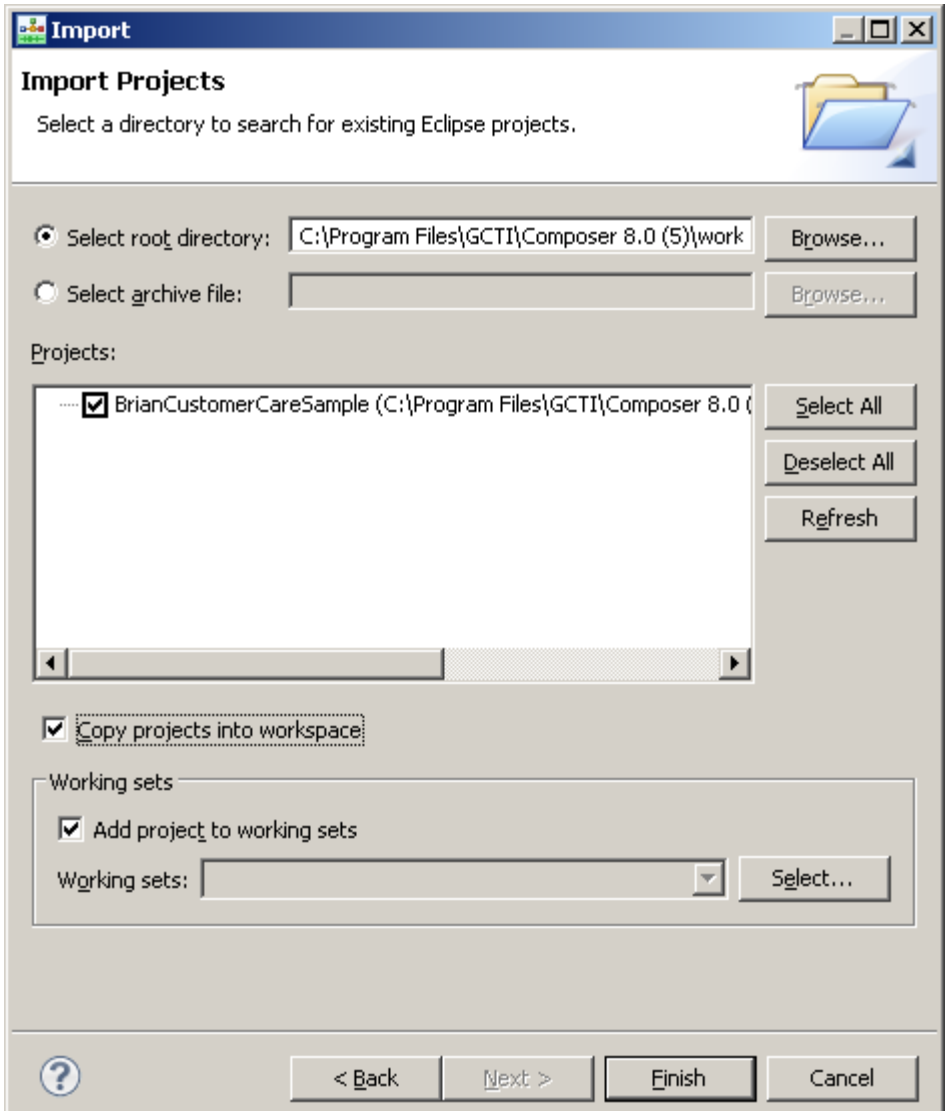

Figure 19: Import Projects Dialog Box

5. Click `Finish`.
6. In the Project Explorer view, select the imported project and type F5 to refresh.
7. Right-click the imported Project and select `Upgrade Composer Project` from the context menu.

8. If the Project is updated, a message appears indicating that it is the current version. Otherwise, a prompt appears asking if you would like to migrate this project. Click the `Yes` button to start the migration process.
9. View the migration report Once the upgrade process is complete, Composer displays a migration report. The migration report is located in the `upgradeReports` folder of the Project; for example:
   `C:\Work\Temp1\Gate3IPTest\upgradeReports\UpgradeReport_Gate3IPTest20090513155840979.html`

End of procedure

Figure 20 shows an example report.

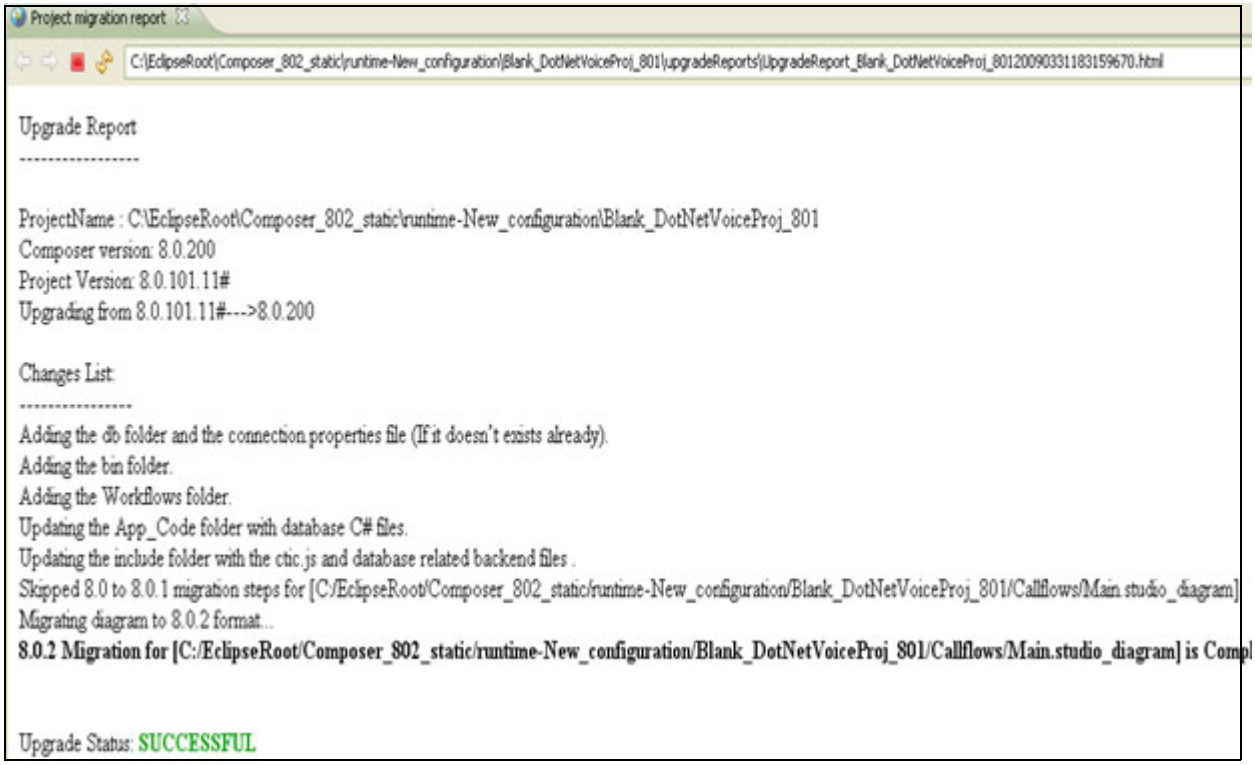

Figure 20: Project Migration Report Example

Migration Error Message

After a Composer Project upgrade, the Project Migration Report may display the following error message:

`error while updating the .studio_config.properties.`

In this case, permissions for .studio_config.properties may be read-only or hidden. To resolve this issue, go to the file system and check for the studio_config.properties file located under the Composer Project directory. Set the file permissions so that the read-only and hidden file attributes are disabled/unchecked.

Hint

To find where the current Project directory is located, do the following:

1. Go to Composer's Project Explorer view.
2. Right-click the Composer Project.
3. From the shortcut menu, select `Properties` > `Resource` and look for Location., e.g., `Location: C:\Program Files\GCTI\Composer 8.0\workspace\JavaComposerProject`

Diagram Level Migration

Note: In Composer 8.0.2 and later, diagrams for voice applications are called "callflow diagrams" whereas in earlier versions of Composer they were called "studio_diagrams."

Procedure: Diagram File Migration

Follow the steps below if you have only copied older diagram files to a current version of Composer Project (or to an already migrated Composer Project).

Start of procedure

1. In Composer's Project Explorer view, select the Project destination folder to where you want the files to be imported (`Callflows` Project folder).
2. Right-click and select `Import`.
3. In the Import wizard, select the diagram files to import.
4. After the import operation completes, right-click on the imported diagram file in the Project Explorer and select `Upgrade Callflow Diagram`.

End of procedure

Notes on Voice Application Migration

It is important to note the following:

- Project migration does not migrate any custom blocks. When Composer is launched, it checks if any custom blocks need migration and migrates them. There are no manual steps involved.
- When migrating from 8.0.2 to 8.0.3/8.0.4, the Entry block variable `_COMPOSER_WSSTUBBING` is renamed `COMPOSER_WSSTUBBING`.

- When migrating from 8.0.1 to 8.0.2, the Studio Diagram file extension changes from `.studio_diagram` to `.callflow`. For example: `MyDiagram.studio_diagram` changes to `MyDiagram.callflow`.

  To avoid any resulting file name conflict, the diagram upgrade will append a timestamp to the file name only if a `.callflow` file with the same file name already exists in the same folder; for example: `Main_2010_02_19_123010.callflow`.

  The Timestamp is of the following format: `yyyy_MM_dd_HHmmss`

  Starting with 8.0.2, the following callflow blocks contain a mandatory Output Result property: Menu, DB Input, Grammar Menu, Input, Get Access Number, Transfer, Statistics and Record. You supply this property by selecting a variable. Since this property is mandatory; if not supplied, an error occurs in the `Problems View` when validating the callflow.

  Migration to 8.0.2 automatically populates this variable. For example, if the block is a Menu block and the block's name is `Main_Menu`, migration will add a `Main_Menu` variable to the Entry block (as if you added it manually) and will set the Output Results property to this variable.
- The GVP Next Generation Interpreter does not support the `error.badfetch.badxmlpage` event. If migrating a callflow application from an earlier version that listed this event under `Supported` in its Entry block `Exceptions` dialog box, you will need to modify that Entry block by removing that event under `Supported` in the `Exceptions` dialog box.

Routing Migration 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. Migrating 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. Migrating workflow diagrams created in the 8.0.3 release of Composer is therefore not supported.
- Composer currently does not support migrating strategies created with Universal Routing 7.x or 8.x into Composer Projects. For assistance in manually porting IRD strategies to SCXML-based strategies, refer to *Universal Routing SCXML, Release 8.0 White Paper*, which is provided as part of the URS 8.0 Release package. The White Paper describes SCXML-equivalent objects and functions.

Deploying Projects to Tomcat

If you already have a project in the workspace and did not perform the Tomcat configuration described earlier, perform the following steps to deploy the project on Tomcat.

Procedure: Deploying Projects to Tomcat

Start of procedure

1. In the `Composer Project Explorer`, right-click the name of the Composer project and select `Properties`.

   The `Properties` window opens.
2. In the left panel, select `Tomcat Deployment`.
3. Click the `Deploy` button to deploy the project.
4. Click `OK` to close the `Properties` window.

End of procedure

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

Related Documentation Resources

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

Composer Help

- *Composer 8.0.4 Help,* which provides integrated help information about using Composer to develop voice and routing applications. Selecting `Help` > `Help Contents` brings up the window shown in Figure 21.

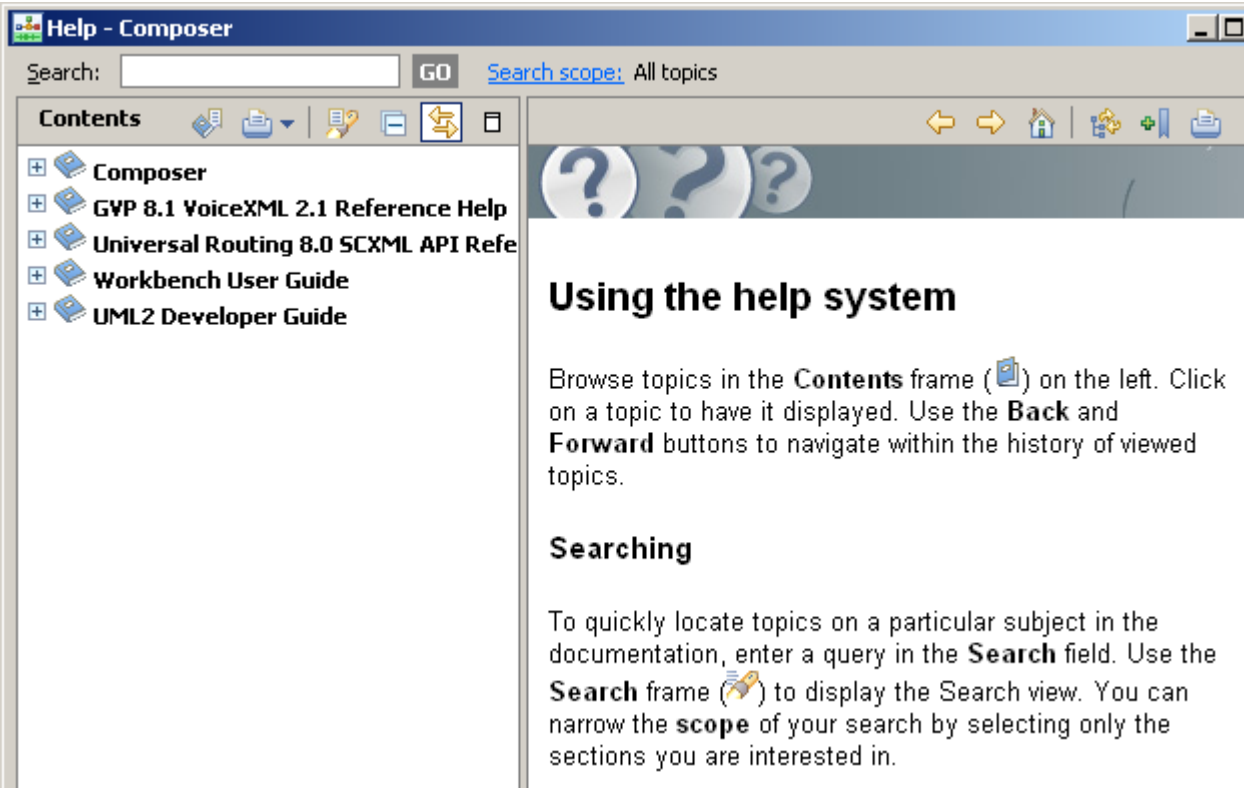

Figure 21: Composer Help Dialog Box

- Double-click Composer.
- Or, since Composer is based on Eclipse, you should familiarize yourself with basic Eclipse concepts by referring to the *Workbench User Guide* shown in Figure 21.

Selecting `Help` > `Search` opens a search pane on the right side of the Composer window. Figure 22 shows example search results.

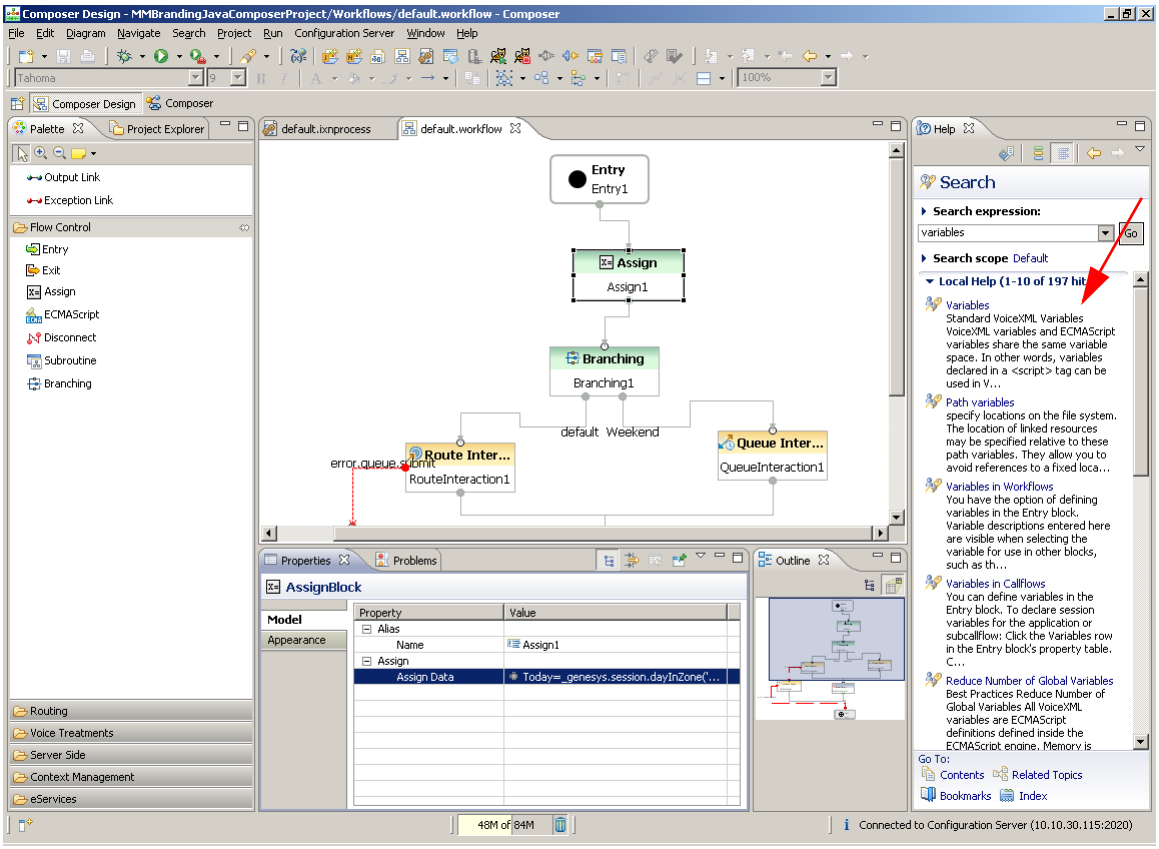

Figure 22: Help Tab for Searching

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 application (see Figure 23).

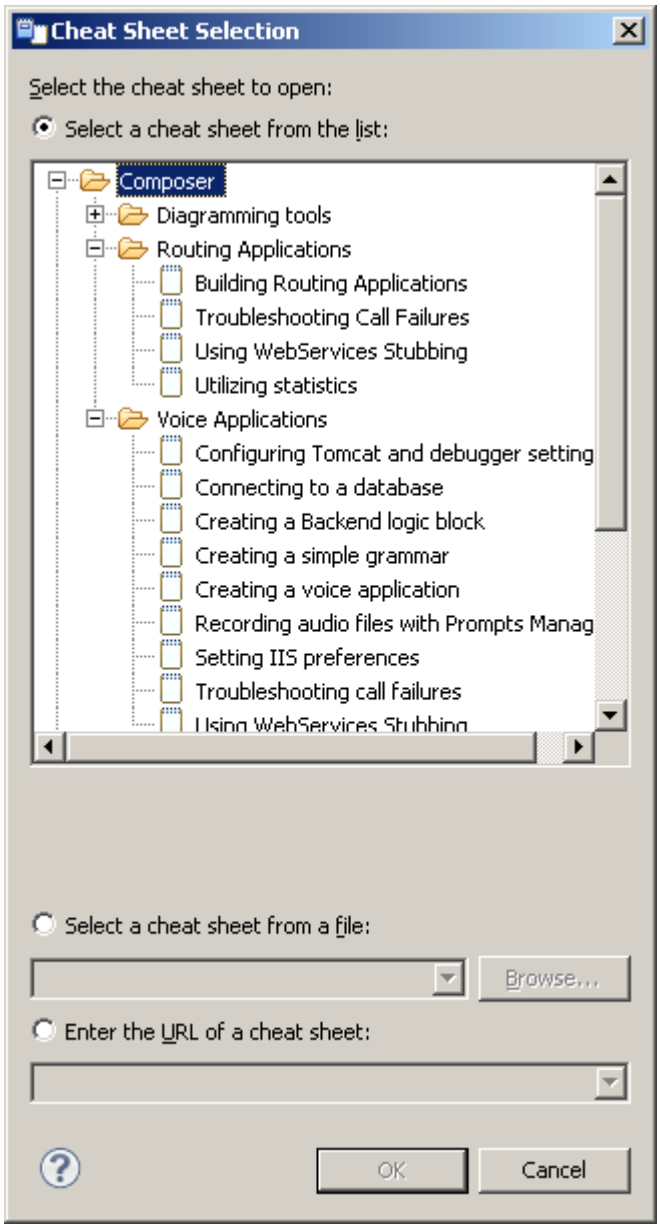

Figure 23: Cheat Sheets

Management Framework

- *Framework 8.0 Deployment Guide,* which provides information about configuring, installing, starting, and stopping Framework components.
- *Framework 8.0 Genesys Administrator Help,* which provides information about configuring and provisioning contact center objects by using the Genesys Administrator.
- *Framework 8.0 Configuration Options Reference Manual,* which provides descriptions of the configuration options for Framework components.

- *Framework 8.0 Stat Server User's Guide*, which describes the configuration, installation, and start procedures relevant to deploying Stat Server.

SIP Server

- *Framework 8.0 SIP Server Deployment Guide,* which provides information about configuring and installing SIP Server.

Universal Routing

- *Universal Routing 8.0 SCXML API Reference*. This help file, available via **Help** > **Contents** (see Figure 21 on page 75), groups URS extensions to the SCXML executable content by Genesys-supplied Functional Module.
- *Orchestration Server 8.0 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

- *Universal Contact Server 8.0 Context Services User's Guide*, 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 Genesys VoiceXML 2.1 Reference Help* (see Figure 21 on page 75), 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.
- *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 8.1, SIP Server 8.0, 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 Technical Support website in the following documents:

- *Genesys Supported Operating Environment Reference Manual*
- *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 7.6 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_v8.0.001.00

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 82).	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

Symbols

A

B

C

D

E

F

G

H

I

M

N

O

P

R

S

T

U

V

W

X