Interaction SDK 7.6

Java

Deployment Guide

About Genesys

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

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

Notice

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

Your Responsibility for Your System's Security

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

Trademarks

Genesys and the Genesys logo are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other company names and logos may be trademarks or registered trademarks of their respective holders. © 2013 Genesys Telecommunications Laboratories, Inc. All rights reserved.

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 14. For complete contact information and procedures, refer to the ***Genesys Care Program 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: 76sdk_dep_ixn_java_07-2013_v7.6.502.00

Table of Contents

List of Procedures

Preface

Welcome to the *Interaction SDK 7.6 Java Deployment Guide*. This document introduces you to the concepts, terminology, and procedures relevant to deploying this Genesys solution. This document is valid only for the 7.6.*x* release(s) of this product.

Note: For releases of this document created for other releases of this product, please 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 these sections:

The Interaction SDK Set

Interaction SDK Java comprises three core application programming interfaces (APIs):

- Agent Interaction (Java API), an object-based library interface.
- Media Interaction (Java API).
- Queued Interaction (Java API).

These components allow the processing of voice, multimedia, and Open Media interactions. Using these APIs, you can integrate virtually all activities into your contact center, queue these interactions, and present the interactions to agents based on priorities that you set in your business rules.

Note: Below, Agent Interaction (Java API) is sometimes abbreviated *AIL*.
Media Interaction (Java API) is abbreviated *MIL*.
Queued Interaction (Java API) is abbreviated *QIL*.

Agent Interaction (Java API) is a set of components that interact with Genesys servers. AIL exposes an interface that enables you to develop agent applications that follow any distributed architecture (stand-alone, *N*-Tier, and so on). With AIL, you can control and manage interactions for many different kinds of media including:

- E-mail
- Chat
- Cobrowse
- Voice (including Callback and Outbound Campaign calls)
- Open Media

MIL enables your applications to process Open Media interaction types, such as those associated with:

- Fax servers
- Workflow systems
- Non-Genesys e-mail management and web chat applications
- Scanned documents
- Web-based training
- Short Message Service (SMS)
- Media types that you define

QIL enables your applications to monitor and manage multimedia and Open Media interactions.

This guide outlines how to configure and install AIL, MIL, and QIL, using the Installation Package CD. That Installation Package includes the following source files:

- AIL 7.6, MIL 7.6, and QIL 7.6 APIs for all supported operating systems.
- Agent Interaction SDK Client template
- Agent Interaction SDK Server template
- Media Interaction SDK template
- Queued Interaction SDK template

Intended Audience

This document, primarily intended for anyone configuring and installing Interaction SDK 7.6 Java, assumes that you have a basic understanding of:

- Computer-telephony integration concepts, processes, terminology, and applications.
- Network design and operation.
- Java programming.
- Your own network configurations.

You should also be familiar with the use of Genesys Configuration Manager and the Genesys Framework.

Usage Guidelines

The Genesys developer materials outlined in this document are intended to be used for the following purposes:

- Creation of contact-center agent desktop applications associated with Genesys software implementations.
- Server-side integration between Genesys software and third-party software.
- Creation of a specialized client application specific to customer needs.

The Genesys software functions available for development are clearly documented. No undocumented functionality is to be utilized without Genesys's express written consent.

The following Use Conditions apply in all cases for developers employing the Genesys developer materials outlined in this document:

1. Possession of interface documentation does not imply a right to use by a third party. Genesys conditions for use, as outlined below or in the *Genesys Developer Program Guide,* must be met.
2. This interface shall not be used unless the developer is a member in good standing of the Genesys Interacts program or has a valid Master Software License and Services Agreement with Genesys.
3. A developer shall not be entitled to use any licenses granted hereunder unless the developer's organization has met or obtained all prerequisite licensing and software as set out by Genesys.
4. A developer shall not be entitled to use any licenses granted hereunder if the developer's organization is delinquent in any payments or amounts owed to Genesys.

5. A developer shall not use the Genesys developer materials outlined in this document for any general application development purposes that are not associated with the above-mentioned intended purposes for the use of the Genesys developer materials outlined in this document.
6. A developer shall disclose the developer materials outlined in this document only to those employees who have a direct need to create, debug, and/or test one or more participant-specific objects and/or software files that access, communicate, or interoperate with the Genesys API.
7. The developed works and Genesys software running in conjunction with one another (hereinafter referred to together as the "integrated solutions") should not compromise data integrity. For example, if both the Genesys software and the integrated solutions can modify the same data, then modifications by either product must not circumvent the other product's data integrity rules. In addition, the integration should not cause duplicate copies of data to exist in both participant and Genesys databases, unless it can be assured that data modifications propagate all copies within the time required by typical users.
8. The integrated solutions shall not compromise data or application security, access, or visibility restrictions that are enforced by either the Genesys software or the developed works.
9. The integrated solutions shall conform to design and implementation guidelines and restrictions described in the *Genesys Developer Program Guide* and Genesys software documentation. For example:
    a. The integration must use only published interfaces to access Genesys data.
    b. The integration shall not modify data in Genesys database tables directly using SQL.
    c. The integration shall not introduce database triggers or stored procedures that operate on Genesys database tables.

Any schema extension to Genesys database tables must be carried out using Genesys Developer software through documented methods and features.

The Genesys developer materials outlined in this document are not intended to be used for the creation of any product with functionality comparable to any Genesys products, including products similar or substantially similar to Genesys's current general-availability, beta, and announced products.

Any attempt to use the Genesys developer materials outlined in this document or any Genesys Developer software contrary to this clause shall be deemed a material breach with immediate termination of this addendum, and Genesys shall be entitled to seek to protect its interests, including but not limited to, preliminary and permanent injunctive relief, as well as money damages.

Chapter Summaries

In addition to this preface, this document contains the following chapters:

- Chapter 1, “AIL, MIL, and QIL Basics,” on page 15, provides information about which systems you must set up before AIL, MIL, and QIL installation and configuration.
- Chapter 2, “Installing and Configuring Interaction SDK Components,” on page 19 indicates how to install AIL, MIL, and QIL via the Genesys Interaction SDK Installation Package CD.
- Chapter 3, “Custom Applications,” on page 27, gives details about installing and configuring your custom built application to run in a Genesys environment.
- Appendix A, “The Properties Tab,” on page 37, provides the information needed to configure the properties tab of each Genesys Interaction SDK component.
- Appendix B, “Switch-Specific Support Configuration,” on page 47, provides the information for switch-specific configuration.
- Appendix C, “Locating your SDK Component’s Files,” on page 51, gives details on where to locate your required files for your SDK component.
- Appendix D, “Configuring the Options Tab,” on page 53, provides the information needed to configure the options tab of each Genesys Interaction SDK component.

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:

76sdk_dep_ixn_java_11-2007_v7.6.000.02

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

Type Styles

Italic

In this document, italic is used for emphasis, for documents' titles, for definitions of (or first references to) unfamiliar terms, and for mathematical variables.

Examples:

- Please consult the *Genesys Migration Guide* for more information.
- *A customary and usual practice* is one that is widely accepted and used within a particular industry or profession.
- Do *not* use this value for this option.
- The formula, $x + 1 = 7$ where x stands for . . .

Monospace Font

A monospace font, which looks like `teletype or typewriter text`, is used for 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; and code samples.

Examples:

- Select the `Show variables on screen` check box.
- Click the `Summation` button.
- In the `Properties` dialog box, enter the value for the host server in your environment.
- In the `Operand` text box, enter your formula.
- Click `OK` to exit the `Properties` dialog box.
- The following table presents the complete set of error messages T-Server® distributes 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.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

Example:

- Enter `exit` on the command line.

Screen Captures Used in This Document

Screen captures from the product GUI (graphical user interface), as used in this document, may sometimes contain a minor spelling, capitalization, or grammatical error. 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.

Square Brackets

Square brackets indicate that a particular parameter or value is optional within a logical argument, a command, or some programming syntax. That is, the parameter's or value's presence is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information. Here is a sample:

```
smcp_server -host [/flags]
```

Angle Brackets

Angle brackets indicate a placeholder for a value that the user must specify. This might be a DN or port number specific to your enterprise. Here is a sample:

```
smcp_server -host <confighost>
```

Related Resources

Consult these additional resources as necessary:

- *Agent Interaction SDK 7.6 Java Developer's Guide*
- *Media Interaction SDK 7.6 Java Developer's Guide*
- *Queued Interaction SDK 7.6 Java Developer's Guide*
- Documentation on Apache Foundation Java-platform components is available on the Apache website at `http://jakarta.apache.org`.
- Documentation and tutorials on Java APIs are available onthe Sun website at `http://java.sun.com`.
- *Multimedia 7.6 Event Media Deployment Guide*, which introduces you to the architecture, required components, and procedures relevant to the deployment of a Genesys Multi-Channel Routing solution.
- *Genesys Events and Models Reference Manual*
- *Genesys Master Glossary*, which ships on the Genesys Documentation Library DVD, provides a fairly comprehensive list of Genesys and CTI terminology and acronyms.
- The Release Notes and Product Advisories for this product, which are available on the Genesys Documentation website at `http://docs.genesyslab.com.`

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

- *Genesys Supported Operating Systems and Databases*
- *Genesys Supported Media Interfaces*

Genesys product documentation is available on the:

- Genesys Technical Support website at http://genesyslab.com/support.
- Genesys Documentation website at http://docs.genesyslab.com.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

Making Comments on This Document

If you especially like or dislike anything about this document, please 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 information in this document only and to the way in which the information is presented. Speak to Genesys Technical Support if you have suggestions about the product itself.

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

Contacting Genesys Technical Support

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

Chapter

1 AIL, MIL, and QIL Basics

This chapter provides an overview of the three application program interfaces (APIs) whose installation is covered individually in the chapters that follow: Agent Interaction (Java API) 7.6, Media Interaction (Java API) 7.6, and Queued Interaction (Java API) 7.6. The chapter contains the following sections:

API Roles

To plan your deployment of Interaction SDK Java, please start by ensuring that you have a supported operating system and Java environment. Refer to the *Genesys Supported Operating Environment Reference Guide* document. (For access details, see "Related Resources" on page 13.)

Agent Interaction (Java API)

Agent Interaction (Java API), also known as AIL, lets you build Java applications to control and manage voice, multimedia, and Open Media interactions issued by, or intended for, a contact center agent.

Media Interaction (Java API)

Media Interaction (Java API), also known as MIL, lets you build Java applications to manage Open Media interactions in the Genesys Framework. MIL 7.6 provides a simple Java API that includes manager interfaces for developing server applications that:

- Create and manage Open Media interactions submitted to Interaction Server.
- Manage Open Media interactions submitted to the Universal Contact Server's database.
- Use ESP protocol to handle interactions' extensions through Interaction Server.

- Monitor your application run mode from the Local Control Agent component point of view.

Note: MIL does not support the Genesys Chat or E-mail media types.

Queued Interaction (Java API)

Queued Interaction (Java API), also known as QIL, lets you build Java applications to control queues made available by the Genesys Framework. QIL 7.6 provides a simple Java API, including manager interfaces to develop applications that can be used:

- To get business attributes and their values.
- To monitor changes in the queues' state, and in associated interactions.
- To get events on interactions in the queues.
- For Genesys media types, and for Open Media types.

QIL 7.6 offers the ability to perform the following ad-hoc management transactions:

- `stopProcessing`
- `placeInQueue`
- `lock/unlock`
- `set/RemoveProperties`
- `pull/leave`

QIL 7.6 retrieves queue content from Interaction Server, in order to provide the status of queue content to QIL's clients.

Components

The main components of each SDK layer are described below:

- "Agent Interaction (Java API)" on page 16
- "Media Interaction (Java API)" on page 17
- "Queued Interaction (Java API)" on page 17

Agent Interaction (Java API)

AIL is comprised of the following components:

- The Agent Interaction (Java API) library, which is written entirely in the Java language and delivered as a set of `.jar` files.
- A javadoc API reference, which is an HTML tree in the `docs/` directory of the installed product directory tree.
- A developer's guide, which is delivered on the documentation CD.
- A set of code examples that exercise some important features of the API, delivered in `.zip` and `.tar.gz` format on the documentation CD.

Media Interaction (Java API)

MIL is comprised of the following components:

- The Open Media common library, written entirely in the Java language and delivered as a set of `.jar` files on the product CD.
- The Media Interaction (Java API) library, which is written entirely in the Java language and delivered as a set of `.jar` files on the product CD.
- A javadoc API reference, which is an HTML tree in the `docs/` directory of the installed product directory tree.
- A developer's guide, which is delivered on the documentation CD.
- A set of code examples that exercise some important features of the API, delivered in `.zip` and `.tar.gz` format on the documentation CD.

Queued Interaction (Java API)

QIL is comprised of the following components:

- The Queued Interaction (Java API) library, which is written entirely in the Java language and delivered as a set of `.jar` files on the product CD.
- A javadoc API reference, which is an HTML tree in the `docs/` directory of the installed product directory tree.
- A developer's guide, which is delivered on the documentation CD.
- A set of code examples that exercise some important features of the API, delivered in `.zip` and `.tar.gz` format on the documentation CD.

Tasks and Their Related Procedures

Table 1 on page 17 summarizes the tasks and their related procedures addressed in this guide.

Table 1: Tasks and Related Procedures

Objective	Related Procedures and Actions
To set up your environment so that you can develop using your AIL Interaction SDK component.	Preparing to Configure your Interaction SDK Component, page 20 Configuring your Interaction SDK for Deployment, page 22 "Configuring the Properties Tab" on page 37 "Configuring the Options Tab for AIL" on page 53

Table 1: Tasks and Related Procedures (Continued)

Objective	Related Procedures and Actions
To install your AIL Interaction SDK component for use in developing custom applications.	Launching System Installation, page 23 Installing your Interaction SDK Component, page 24
To set up your environment so that you can develop using your QIL Interaction SDK component.	Preparing to Configure your Interaction SDK Component, page 20 Configuring your Interaction SDK for Deployment, page 22 "Configuring the Properties Tab" on page 37 "Configuring the Options Tab for QIL" on page 66
To install your QIL Interaction SDK component for use in developing custom applications.	Launching System Installation, page 23 Installing your Interaction SDK Component, page 24
To set up your environment so that you can develop using your MIL Interaction SDK component.	Preparing to Configure your Interaction SDK Component, page 20 Configuring your Interaction SDK for Deployment, page 22 "Configuring the Properties Tab" on page 37 "Non-mandatory." on page 68
To install your MIL Interaction SDK component for use in developing custom applications.	Launching System Installation, page 23 Installing your Interaction SDK Component, page 24
To allow your custom application to run in a Genesys environment.	Setting up VoIP Support, page 27 Setting up your SIP Communication Server, page 28 Configuring for your Specific Switch, page 30 Running MIL and QIL on the Same JVM, page 31 Starting MIL in Server Mode, page 31 Configuring External Service Protocol Request, page 33

Chapter

2 Installing and Configuring Interaction SDK Components

This chapter describes how to configure and install the Agent Interaction (Java API), Multimedia Interaction (Java API), and Queued Interaction (Java API) components in your Genesys environment, so that you can develop custom applications. These components will be referred to as AIL, MIL, and QIL. The chapter contains the following sections:

Note: Open Media services have been extended to support ad-hoc management. The extensions include methods enabling a supervisor to perform queue content modifications—such as retrieving information after monitoring the queue, and routing interactions from Interaction Server to agents.

Configuration and Installation

There are two main procedures that must be followed to complete the task of configuring your Interaction SDK component:

- "Preparing to Configure your Interaction SDK Component" on page 20
- "Configuring your Interaction SDK for Deployment" on page 22

If your required template is not present in Configuration Manager you must also complete the procedure "Importing an Application Template into Configuration Manager" on page 21.

Two procedures must be completed to install your Interaction SDK component:

- "Launching System Installation" on page 23
- "Installing your Interaction SDK Component" on page 24

Procedure: Preparing to Configure your Interaction SDK Component

Purpose: To choose your configuration environment and verify that you have the required application templates before you configure your Interaction SDK component to work in your Genesys environment.

Prerequisites

- If the template is not present, you will have to import the application template into Configuration Manager. For information of how to import an application template, see "Importing an Application Template into Configuration Manager" on page 21.

Start of procedure

1. Choose your configuration environment. Depending on which of the Interaction SDKs that you choose to install, your component might have configuration environment alternatives. For details about making this choice, refer to the *Agent Interaction SDK 7.6 Java Developer's Guide*. You can choose between the alternatives below.
   - AIL:
     - Stand-Alone (Interaction SDK client)
     - *N*-Tier (Interaction SDK server)
   - MIL:
     - Stand-Alone
   - QIL:
     - Stand-Alone
2. Verify the template. Before creating an application, check to see that an application template exists. The template provides most of the application's configuration options and default values.
   a. In Configuration Manager, open the `Environment` folder, and then open the `Application Templates` folder.
   b. Select the template for your Interaction SDK component:
      - AIL:
        - `Agent_Interaction_client_765.apd`
        - `Agent_Interaction_server_765.apd`

- MIL:
  - `Media_Interaction_SDK_server_765.apd`
- QIL:
  - `Queued_Interaction_server_765.apd`

End of procedure

Next Steps

- You may need to import an application template into Configuration Manager using Importing an Application Template into Configuration Manager, page 21.

Procedure: Importing an Application Template into Configuration Manager

Purpose: To import the required application templates before you configure your Interaction SDK component to work in your Genesys environment.

Prerequisites

- Review the *Agent Interaction SDK Java Developer's Guide for* configuration environment alternatives.

Start of procedure

1. Open the `Environment` folder and select the `Application Templates` folder.
2. From the `File` menu, select `Import Application Template`.
3. In the `Open` window that appears, import one of the following templates from your CD:
   - AIL:
     - `templates/Agent_Interaction_client_765.apd`
     - `templates/Agent_Interaction_server_765.apd`
   - MIL:
     - `templates/Media_Interaction_server_765.apd`
   - QIL:
     - `templates/Queued_Interaction_server_765.apd`
4. Click `Open`. The corresponding `Properties` window opens.
5. Click `OK`. The template is imported into the `Application Templates` folder.

6. Verify that the host on which you will install the component's server is declared in the `Hosts` folder.

End of procedure

Next Steps

- You will want to configure your Interaction SDK using Configuring your Interaction SDK for Deployment, page 22.

Procedure: Configuring your Interaction SDK for Deployment

Purpose: To configure your Interaction SDK component to work in your Genesys environment.

Prerequisites

- Before beginning the configuration process, you must import the template. To do so, follow the steps in the Importing an Application Template into Configuration Manager, page 21.

Start of procedure

1. In Configuration Manager, open the `Environment` folder and select the `Applications` folder.
2. From the `File` menu, select `New > Application`. The Browse window opens, listing all application templates present in Configuration Manager.
3. Choose the application template.
   - AIL:
     - `Agent_Interaction_server_765.apd`
     - `Agent_Interaction_client_765.apd`.
   - MIL:
     - `Media_Interaction_SDK_server_765.apd`
   - QIL:
     - `Queued_Interaction_server_765.apd`
4. Click `OK`. The `Properties` window appears.
5. Configure the various tabs in the `Properties` window. For information of how to configure these tabs, see Appendix A, "The Properties Tab," on page 37.

End of procedure

Next Steps

- You will need to configure the `properties` tab. Follow the guidelines in Appendix A, "The Properties Tab," on page 37.
- You will need to configure the `options` tab. Follow the guidelines in Appendix D, "Configuring the Options Tab," on page 53.

Procedure: Launching System Installation

Purpose: To locate and launch the your Interaction SDK's installation wizard.

Prerequisites

- Be sure that your target machine meets the platform and component prerequisites listed in the *Genesys Supported Operating Environment Reference Guide* document. (For access details, see "Related Resources" on page 13.)
- Before installing on a Red Hat Enterprise Linux 6 64-bit operating system, you must first install the Red Hat compatibility packages.
- Locate the compressed installation files, readme file, and templates on the installation CD-ROM. The information can be found below:
  - AIL: See "Locating the AIL Component's Files" on page 51.
  - MIL: See "Locating the MIL Component's Files" on page 52.
  - QIL: See "Locating the QIL Component's Files" on page 52.

Windows Installation

Start of procedure

1. Navigate to the component's directory on the CD.
   - AIL:
     - `AgentInteraction\Java\windows\`
   - MIL:
     - `MediaInteraction\Java\windows\`
   - QIL:
     - `QueuedInteraction\Java\windows\`
2. Run `Setup.exe`. Note the installation's default location.
   - AIL:
     - `Program Files\GCTI\Interaction SDK for Java 7.6\AIL\`
   - MIL:
     - `Program Files\GCTI\Interaction SDK for Java 7.6\MIL\`
   - QIL:

- Program Files\GCTI\Interaction SDK for Java 7.6\QIL\

3. Follow the onscreen instructions and prompts to complete the installation.

End of procedure

Next Steps

- You can now install your component using Installing your Interaction SDK Component, page 24.

UNIX Installation

Prerequisites

- Check to make sure you have installation rights.

Start of procedure

1. Navigate to the component's directory on the CD.
   - AIL:
     - AgentInteraction/Java/<os>/
   - MIL:
     - MediaInteraction/Java/<os>/
   - QIL:
     - QueuedInteraction/Java/<os>/
2. Run the `install.sh` script.
3. Follow the onscreen instructions and prompts to complete the installation.

Note: After running the Windows or UNIX executable or script, you should inspect the directory tree to make sure your installation is consistent with the installation you chose.

End of procedure

Next Steps

- You can now install your component using Installing your Interaction SDK Component, page 24.

Procedure: Installing your Interaction SDK Component

Purpose: To install your Interaction (Java API) 7.6 component:

Prerequisites

- You must have completed the steps in Launching System Installation, page 23.

Start of procedure

1. Launch the install script or executable corresponding to your host operating system.
2. If an `Open File — Security Warning` message box appears, click its `Run` button to proceed with installation.
3. Follow the prompts to choose:
   - documentation (javadoc) only
   - libraries (`.jar` files) only
   - both documentation and libraries
4. Enter the destination locations.

   Table 2 on page 25 lists the libraries installed during system installation.

Table 2: Libraries Installed During System Installation

Libraries	Vendor
Javamail	java.sun.com
Java Beans Activation Framework	java.sun.com
Log4j	jakarta.apache.org
Xerces_J, including: • Dom interface • SAX interface • Jaxp	jakarta.apache.org; • w3c.org • www.saxproject.org • java.sun.com

End of procedure

Next Steps

- You can now use your Interaction SDK component to create custom built applications. For examples see Chapter 3, “Custom Applications,” on page 27.

Chapter

3 Custom Applications

This chapter describes how to install and configure your custom built applications to run in a Genesys environment. The chapter contains the following sections:

Install and Configure Custom Applications

There are many ways to customize your Genesys environment to meet your business needs. Below are a few examples of how to install and configure your Interaction SDK custom built applications.

Procedure: Setting up VoIP Support

Purpose: To allow your call center to use VoIP technology by creating and configuring a DN on the IPMX (Genesys IP Media eXchange) switch.

Prerequisites

- You must have your Interaction SDK component installed. For detailed information on how to install Interaction SDK components, see Chapter 2, “Installing and Configuring Interaction SDK Components,” on page 19.

- To handle VoIP calls, Agent Interaction (Java API) uses H323 (Microsoft NetMeeting) or SIP (Windows Messenger) applications. For a full description of VoIP, see the *Genesys IP Media eXchange Reference Guide.*

Start of procedure

1. General Tab
   - `Type: ACDPosition`
   - `Number: any`
2. Advanced Tab
   - `Alias` and/or `Use-Override` (must be selected if used): Enter the IP address on which the agent's VoIP application is launched, preceded by the @ symbol and followed by the `/VR` option.

     For example: `@192.168.3.4/VR`

     This option allows a Connected mode between IPMX and a VoIP application. For standard use, Genesys recommends setting the agent's NetMeeting to `autoresponse`.

`Note:` Agents can receive VoIP sessions when logged into an IPMX DN. The connection between VoIP applications and IPMX is established only after the agent accepts the call, and the connection remains on until the agent logs out.

End of procedure

Next Steps

- Customize your call center further by:
  - Setting up your SIP Communication Server, page 28.
  - Configuring for your Specific Switch, page 30.
  - Running MIL and QIL on the Same JVM, page 31.
  - Starting MIL in Server Mode, page 31.
  - Configuring External Service Protocol Request, page 33.

Procedure: Setting up your SIP Communication Server

Purpose: To allow your call center to use SIP communication technology.

Agent Interaction (Java API) supports SIP Communication Server. For a full description of this server, see the *Framework SIP Communication Server Deployment Guide.*

You configure endpoints (SIP phones) as DN objects of type Extension in the Genesys Configuration Layer. For DN objects, you must configure the Configuration Manager fields listed in the following subsections.

Prerequisites

- You must have your Interaction SDK component installed. For detailed information on how to install Interaction SDK components, see Chapter 2, "Installing and Configuring Interaction SDK Components," on page 19.

Start of procedure

You configure endpoints (SIP phones) as DN objects of type Extension in the Genesys Configuration Layer. For DN objects, you must configure the Configuration Manager fields listed in the following subsections.

1. General Tab
   - Type: `Extension`
   - `Number`: The username part of the endpoint's Address of Record (AOR). Contains a numeric-only DN number that can be dialed directly from a phone.

Note: You must not use an @ sign or a domain name when configuring the number.

---

2. Annex Tab
   - `Annex TServer/contact`: Contains the contact URI. This field is used to specify the endpoint IP address if this is a fixed address. This object is necessary only for a stand-alone configuration, and only if the endpoint does not register itself in the Communication Server registry.

     The URI format is:

     `[sip:][number@]hostport[;transport={tcp|udp}]`

     Where:
     - `sip` is an optional prefix.
     - `number` is the DN number. The current version of SIP Communication Server ignores this value.
     - `hostport` is a `<host>:<port>` pair, where `<host>` is either a dotted IP address or a DNS-resolvable hostname for the endpoint.
     - `transport=tcp` or `transport=udp` is used to select the network transport.

Note: The realm for password authentication is configured globally. There is one realm per Communication Server.

End of procedure

Next Steps

- Customize your call center further by:
  - Setting up VoIP Support, page 27.
  - Configuring for your Specific Switch, page 30.
  - Running MIL and QIL on the Same JVM, page 31.
  - Starting MIL in Server Mode, page 31.
  - Configuring External Service Protocol Request, page 33.

Procedure: Configuring for your Specific Switch

Purpose: To configure your specific switch.

Prerequisites

- You must have your Interaction SDK component installed. For detailed information on how to install Interaction SDK components, see Chapter 2, "Installing and Configuring Interaction SDK Components," on page 19.

Start of procedure

1. For full support of your specific switch, configure the `Place` that the agent will log into. The `Place` configuration must not be changed while an agent is logged in.
2. Check with your switch to determine the kind of DN support you have. Then configure it accordingly. See, Appendix B, "Switch-Specific Support Configuration," on page 47 for details.

Note: In some of the switches, the agent cannot see all the DNs in the `Place` configuration. In these cases, only one DN is visible, and this DN includes the features of all other DNs.

End of procedure

Next Steps

- Customize your call center further by:
  - Setting up VoIP Support, page 27.
  - Setting up your SIP Communication Server, page 28.
  - Running MIL and QIL on the Same JVM, page 31.
  - Starting MIL in Server Mode, page 31.
  - Configuring External Service Protocol Request, page 33.

Procedure:
Running MIL and QIL on the Same JVM

Purpose: To configure MIL and QIL to run on the same JVM.

Prerequisites

- You must have your Interaction SDK component installed. For detailed information on how to install Interaction SDK components, see Chapter 2, "Installing and Configuring Interaction SDK Components," on page 19.

Start of procedure

In order to run QIL and MIL on the same JVM with the same application in Configuration Manager, no special configuration is needed, because QIL options are a subset of the MIL options.

Note: Genesys advises you to use the MIL template and complete it with the applicable QIL options.

End of procedure

Next Steps

- Customize your call center further by:
  - Setting up VoIP Support, page 27.
  - Setting up your SIP Communication Server, page 28.
  - Configuring for your Specific Switch, page 30.
  - Starting MIL in Server Mode, page 31.
  - Configuring External Service Protocol Request, page 33.

Procedure:
Starting MIL in Server Mode

Purpose: To start MIL in server mode.

Prerequisites

- You must have your Interaction SDK component installed. For detailed information on how to install Interaction SDK components, see Chapter 2, "Installing and Configuring Interaction SDK Components," on page 19.

Start of procedure

1. Launch `startMediaServer.cmd` from the installation directory. The Class used to start MIL in server mode (located in the `MIL.jar`) is `com.genesyslab.omsdk.mil.MILBootstrapper`.
2. In the Command window, add to `startMediaServer.cmd` the following mandatory parameters:
   - `host <config server host>`
   - `port <config server port>`
   - `app <application name>`

   Command line: `startMediaServer -host <config server host> -port <config server port> -app <application name>`.
3. You can add non-mandatory parameters for OMSDKConnector initialization:
   - `backupHost <backup configuration server host name>`
   - `backupPort <backup configuration server port>`
   - `reconnectionPeriod <period between reconnection attempts in milliseconds>`
   - `reconnectionAttempts <number of reconnection attempts>`
4. You can also add the non-mandatory parameter for MIL initialization:
   - `ext <comma-separated list of preloaded extensions (FQN)>`.

   To correctly start MIL in server mode, the Java class path should include all the following `.jars`:
   a. Genesys libraries:
      - Java Config Library (`cfglib.jar`)
      - Java Common Library (`commonLib.jar`)
      - Java TKVList library (`tkv.jar`)
      - Java XKVList library (`xkv.jar`)
      - Java Management Layer Library (`gml.jar`)
      - License library used by Java Config Library (`license.jar`)

   b. Multimedia libraries:
      - ESP Protocol Library Java (`_3rd_party_protocol.jar`)
      - Interaction Server API Java (`_workflow_engine_protocol.jar`)

   c. MIL libraries:
      - OMSDK MIL library (`mil.jar`)
      - om_commons library (`om_commons.jar`)
      - UCS OMAPI library (`omapi.jar`)

   d. Java common libraries:
      - `commons-collections-3.1.jar`
      - `commons-lang-2.0.jar`
      - `concurrent.jar`
      - `log4j-1.2.8.jar`
      - `mail.jar`

e. `.jar` files that contain necessary extensions (if any).

End of procedure

Next Steps

- Customize your call center further by:
  - Setting up VoIP Support, page 27.
  - Setting up your SIP Communication Server, page 28.
  - Configuring for your Specific Switch, page 30.
  - Running MIL and QIL on the Same JVM, page 31.
  - Configuring External Service Protocol Request, page 33.

Procedure: Configuring External Service Protocol Request

Purpose: To define generic blocks in a strategy on which you want to make an External Service Protocol request.

Prerequisites

- You must have your Interaction SDK component installed. For detailed information on how to install Interaction SDK components, see Chapter 2, "Installing and Configuring Interaction SDK Components," on page 19.

Start of procedure

To make requests on Open Media interactions related to your custom media server, you can define generic blocks in your strategy through IRD (Interactive Routing Designer). For further details on Genesys IRD strategies, see the *Genesys Multimedia 7.6 User's Guide*. Define the `External Service` block in a strategy for the Open Media interactions on which you want to make an External Service Protocol (ESP) request. (See Figure 1 on page 34 as a guide to the following instructions.)

1. `Application type`: Select `ThirdPartyServer`.
2. `Application name`: Specify the application (server) name. If you want to use load-balancing of ESP requests, the Universal Routing Server has to call ESP server by type, and the server name should not be specified.
3. `Service`: Specify the Fully Qualified Class Name (FQCN) of the extension class to be used to handle requests.
4. `Method`: Specify the method name to be invoked in the extension class.
5. Specify the parameters to be put into `request parameters` attached to ESP requests.

Note: You can create any key/value pair.

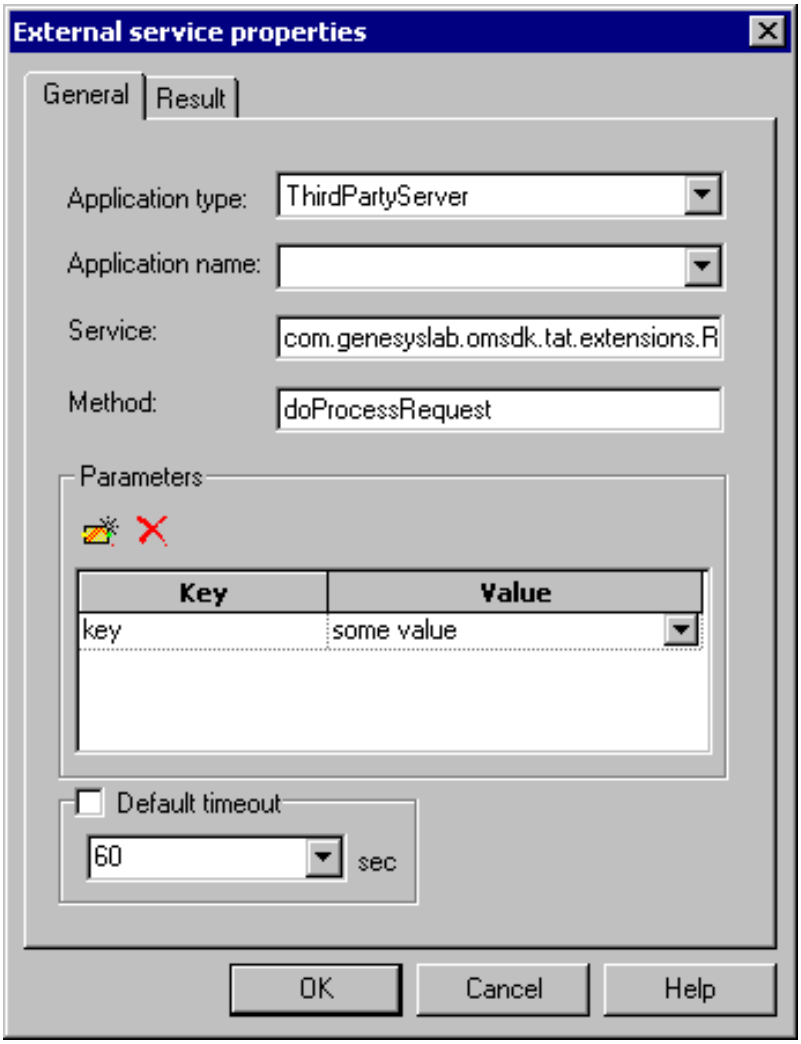

Figure 1: External Service Properties Window

6. `Default timeout`: Specify the timeout for waiting response to ESP request.

Note: You can choose the default value.

7. Click `OK` to save your data.

Note: In order to invoke ESP requests on MIL server, the link to the MIL application should be added to the `Connections` tab in the `IxnServer` application in Configuration Manager.

End of procedure

Next Steps

- Customize your call center further by:
  - Setting up VoIP Support, page 27.
  - Setting up your SIP Communication Server, page 28.
  - Configuring for your Specific Switch, page 30.
  - Running MIL and QIL on the Same JVM, page 31.
  - Starting MIL in Server Mode, page 31.

Appendix

A The Properties Tab

This appendix includes information on how to configure the properties tab of for Agent, Queued, and Media Interaction SDK components. The chapter contains the following sections:

Configuring the Properties Tab

If you create the application in Stand-Alone (Interaction SDK client), no `Server` or `Start info` tabs appear in this window—unlike for an *N*-Tier (Interaction SDK Server) template.

The following sections describe how to configure each tab in the `Properties` window:

General Tab

In the `Name` text box on the `General` tab, enter a name for the application. Below are some examples.

- AIL: `Agent Interaction Layer`
- MIL: `Media Interaction Layer`
- QIL: `Queued Interaction Layer`

Server Info Tab

This tab is typically used with the Genesys Management Layer. For AIL and QIL this tab does not appear.

- MIL: Specify the host where the MIL application resides (this information is used by a low-level LCA service) and the port (used by a low-level ESP service).

Start Info Tab(*N*-Tier Only)

This tab is typically used with the Genesys Management Layer. For AIL, MIL and QIL this tab is not implemented and they do not make use of its data, so you can enter any value in the `Working Directory` and the `Command-Line` text boxes.

- AIL: This tab appears only if you use the `Server` template.

Connections Tab

On the `Connections` tab, add the following basic connections:

- AIL:
  - For Voice-only configuration: `T_Server`.
  - For full configuration: `Universal Contact Server, T_Server, Interaction Server`.
  - For VoIP configuration: select the T-Server of the IPMX T-Server.
  - To configure the ADDP protocol of the configuration server, add a connection to the Configuration Server.
  - To use ADDP protocol for the connections to Configuration Server, T-Server, and Interaction Server, select `ADDP` from the dropdown list of the connection protocol field.
- MIL:
  - Interaction Server (ADDP should be enabled).
  - Universal Contact Server (if low-level UCS service is to be used).
  - MessageServer (if logging into Message Server is to be used).
- QIL:
  - Interaction Server (ADDP should be enabled).
  - MessageServer (if logging into Message Server is to be used).

Options Tab

In Configuration Manager, configuration options are stored within *section* folders. The section folders and options are stored on the `Options` tab; you click a section to bring up its options. The location for detailed information regarding your Interaction SDK's option tab is listed below.

- AIL: See Appendix D, "Configuring the Options Tab for AIL," page 53 as a guide to viewing or changing AIL options.
- MIL: See Appendix D, "Configuring the Options Tab for MIL," page 63 as a guide to viewing or changing MIL options.
- QIL: See Appendix D, "Configuring the Options Tab for QIL," page 66 as a guide to viewing or changing QIL options.

Once all the Options tab is configured:

1. Click `OK` in the `Properties` window.
2. Open Configuration Manager's `Applications` folder.

3. Right-click the Interaction SDK application.
   - AIL: `Agent Interaction (Java API)`
   - QIL: `Queued Interaction (Java API)`
   - MIL: `Media Interaction (Java API)`
4. Select `Properties` to configure the `Security` tab.

Security Tab

This tab is typically used with the Genesys Management layer.

- AIL (Server): In the `Log On As` section, select `SYSTEM` so that AIL will connect to the Configuration Server with `SYSTEM` permissions.
- AIL (Client): There is no `Log On As` section. Check that the user you pass when creating AIL has permission to the application object.
- MIL: In the `Log On As` section, select `SYSTEM` so that MIL will connect to the Configuration Server with `SYSTEM` permissions.
- QIL: In the `Log On As` section, select `SYSTEM` so that QIL will connect to the Configuration Server with `SYSTEM` permissions.

Tenants Tab

The tenants tab applies to multi-tenant configuration only and appears only if you use the `Server` template. The Configuration Server can be installed either as a single or multi-tenant environment; however, AIL, MIL, and QIL only work properly with a single tenant. On this tab, add the single tenant with which your AIL, MIL, or QIL component will work.

- AIL (Server): Set the tenant used (only one tenant) in AIL's `Tenant` tab.
- AIL (Client): There is no `Tenant` tab. Start AIL with a user that belongs to the right tenant.
- MIL: Set the tenant used (only one tenant) in MIL's `Tenant` tab.
- QIL: Set the tenant used (only one tenant) in QIL's `Tenant` tab.

To configure AIL's Tenants Tab for *N*-Tier deployment, see Configuring the Tenants Tab for N-Tier Deployment of AIL using the Standard Method, page 40.

`Note:` The Interaction SDKs are not suited for multi-tenant deployments. Although you can use them for a given tenant in a multi-tenant environment, you would need a separate instance of your application for each tenant using it. (As an alternative, the Genesys Platform SDK supports multi-tenancy.)

Procedure: Configuring the Tenants Tab for N-Tier Deployment of AIL using the Standard Method

Purpose: To configure your AIL Tenants tab for *N*-Tier deployment using the standard method.

If you are setting up a multi-tenant environment, you can also create separate, tenant-specific Persons (accounts) to control the privileges of the Genesys `AgentInteractionSDK` Applications in each tenant. For more information, see Configuring the Tenants Tab for N-Tier Deployment of AIL using Alternative Methods, page 41.

Prerequisites

- You must have the AIL component installed. For detailed information on how to install Interaction SDK components, see Chapter 2, "Installing and Configuring Interaction SDK Components," on page 19.

Start of procedure

To create a new person using the standard method:

1. In Configuration Manager, right-click the `Persons` folder and select `New > Person`.

   If you are working in a single-tenant environment, use the `Persons` folder that is located under `Resources`.

   If you are working in a multi-tenant environment, use the `Persons` folder that is located under the appropriate tenant (see Chapter 3 of this *Deployment Guide* for details).

2. After the `New Person` dialog box appears, click the `General` tab and enter the following parameters:
   - `First:` Agent
   - `Last:` InteractionSDK
   - `Employee ID:` AgentInteractionSDK
   - `User Name:` AgentInteractionSDK
   - `Enter Password:` Unique password
   - `Re-enter Password:` Unique password
   - `State Enabled:` Checked
   - `Is Agent:` Checked

Note: This Person will be enabled to make changes to most Genesys objects managed by the Configuration Server. Ensure that you specify a secure password in order to prevent unauthorized access.

3. Click OK to save your changes.
4. In Configuration Manager, expand the Access Groups navigation tree folder.
   - If you are working in a single-tenant environment, expand the Access Groups folder that is located under Resources.
   - If you are working in a multi-tenant environment, expand the Access Groups folder that is located under the appropriate tenant.
5. Right-click the Administrators navigation tree node and select New > Shortcut to Person.
6. In the Browse dialog box that appears, select the AgentInteractionSDK Person and click OK.

End of procedure

Next Steps

- Configure your component further using the following procedures:
  - Configuring Contact Look Up or Creating Voice Interactions, page 42.

Procedure: Configuring the Tenants Tab for N-Tier Deployment of AIL using Alternative Methods

Purpose: To configure your AIL Tenants tab for *N*-Tier deployment using alternative methods.

If you are setting up a multi-tenant environment, you can also create separate, tenant-specific Persons (accounts) to define the privileges of the Genesys AgentInteractionSDK Applications in each tenant. This section outlines two different methods for creating a Person with tenant-specific privileges. For more information, see the *Configuration Manager online help*.

Prerequisites

- You must have the AIL component installed. For detailed information on how to install Interaction SDK components, see Chapter 2, "Installing and Configuring Interaction SDK Components," on page 19.

Start of procedure

To create a tenant-specific Access Group and Person:

1. Create a new, tenant-specific Access Group called Power Administrators.
2. Grant Full Control privileges to Power Administrators for all objects within the tenant.

3. Create a new, tenant-specific Person called `AgentInteractionSDK`.
4. Add `AgentInteractionSDK` to the `Power Administrators` Access Group.

To create a tenant-specific Person:

1. Create a new, tenant-specific Person called `AgentInteractionSDK`.
2. Grant `Full Control` privileges to `AgentInteractionSDK` for all objects within the tenant.

To associate the Person's account with the Application:

1. In the `Log On As` section, select `This Account`. The `Add User` dialog box appears.
2. Select the appropriate `AgentInteractionSDK` user and click `Add`.
3. Click `OK` to close the `Add User` dialog box.
4. In the `Properties` dialog box, click `OK` to save your changes.

End of procedure

Next Steps

- Configure your component further using the following procedures:
  - Configuring Contact Look Up or Creating Voice Interactions, page 42.

Procedure: Configuring Contact Look Up or Creating Voice Interactions

Purpose: To override the default contact lookup configuration for voice interactions in accordance with your corporate requirements.

By default, AIL attaches a contact to the voice interactions through the customer calling number when Universal Contact Server is connected.

You can customize contact lookup for voice interactions according to your corporate requirements. To override this behavior and search the contact using information in the attached data (sent by IVR), you must configure the following in Configuration Manager.

Prerequisites

- You must have the AIL component installed. For detailed information on how to install Interaction SDK components, see Chapter 2, "Installing and Configuring Interaction SDK Components," on page 19.

Start of procedure

To customize the contact attributes, or to create new ones, for voice interactions in Configuration Manager and in the appropriate tenant you must configure the `Contact Attributes` values in the appropriate tenant. The configuration for Contact Attributes/Attribute Values should follow the descriptions below.

1. Open the `Annex` tab on the tenant object for the tenant where your AIL application is going to run. Create a `settings` section. Then within the `settings` section create the following options and configure them according to your needs.

Settings Section (Annex Tab)

is-searchable

Default Value: `false`
Valid Values: `true, false`

search-order-level

Default Value: `127`
Valid Values: `0` (highest priority), any positive integer up to `127`

is-case-sensitive

Default Value: `false`
Valid Values: `true, false`

Note: The entered option name and value are only examples. It is up to your company to decide which criteria are relevant to your contact lookup.

2. The attributes above must be added for `EmailAddress, PhoneNumber, FirstName,` and `LastName` values.
   - For these four values, the `is-searchable` key is set to `true`.
   - Regarding the `search-order-level`:
     - `EmailAddress=0`
     - `PhoneNumber=1`
     - `FirstName=2`
     - `LastName=2`

   a. In the `Business Attributes` folder, select `Contact Attribute`.
   b. Open the `Attribute Value` folder.
   c. Select `New > Business Attributes` and fill in the following fields:
      - `Name`—Enter the name of the attribute.

Warning! The `voice-attribute` that you create in "The `voice-attribute` section of the AIL application." on page 44 must have the same name as this attribute.

- `Display Name`—Enter the display name you want to give to this attribute.
- `Type`—Select Interaction Operational Attribute.
- `Description`—Describe the attribute, if desired.

d. Right-click on the created `Attribute Value` folder, and select `New > Business Attribute Value`.

e. In the `Attribute Value` folder's `Properties`, fill in the following fields:
- `Name`—Enter the key for the attribute value.
- `Display Name`—Enter the display name of this attribute value.

f. Click `OK`.

g. In the `Attribute Value Properties' Annex` tab, create a `settings` section.

h. Double-click `settings` and click `New`.

i. In the `Edit Option` window, fill in the `Option name` and the `Option value` fields.

j. Set `is-searchable` to `true`.

k. Set `search-order-level` to the required value, according to "search-order-level" on page 43.

3. The `voice-attribute` section of the AIL application.

To customize the contact attributes, or to create new ones, for voice interactions in Configuration Manager and in the appropriate tenant:

To configure the `voice-attribute` option for Voice interactions:

a. In Configuration Manager, select your application's `Properties` window.

b. On the `Options` tab, create a new `voice-attribute section`.

c. In this section, enter the required values (according to your corporate needs) as shown below.

voice-attribute Section

BusinessID

AccountNb

Defines the attribute value that the client application sends to Universal Contact Server. This key is used to get the data from the attached data. The value must fit the business attribute's attribute data name.

Note: The entered option name and value are only examples. It is up to your company to decide which criteria are relevant to your contact lookup.

End of procedure

Next Steps

- Configure your component further using the following procedures:
  - Configuring the Tenants Tab for N-Tier Deployment of AIL using the Standard Method, page 40.
  - Configuring the Tenants Tab for N-Tier Deployment of AIL using Alternative Methods, page 41.

Appendix

B Switch-Specific Support Configuration

For full support of the following switches, configure the `Place` that the agent will log into as described in Table 3.

Table 3: Switch-Specific Support

Switches	DN in Configuration Manager	Agent Login in Configuration Manager	DN ID Reflected
2 DNs (1 Extension and 1 Position)			
Nortel Symposium Nortel Meridian Nortel Communication Server 2000 (formerly DMS100) NEC APEX NEC SV700	2 DNs: • 1 Extension • 1 ACD Position	No constraint	1 Voice DN (ACD Position number)
1 DN or More			
Ericsson MD110	1 DN or more: • 1 Extension (ODN) • *n*= 0/1 ACD Positions (ADN)	No constraint	1 Voice DN (Extension number)

Table 3: Switch-Specific Support (Continued)

Switches	DN in Configuration Manager	Agent Login in Configuration Manager	DN ID Reflected
1 DN (1 Extension or 1 Position)			
Alcatel A4200 emulated Aspect Call Center Avaya Definity G3 Cisco CallManager Dharma EADS Telecom M6500 EADS (Intecom) E EADS (Intecom) Point Span Fujitsu F9600 Mitel SX-2000 Mitel MN-3300 Philips Sopho iS3000 Siemens HiPath 3000 Siemens Realitis-DX iCCL Rockwell Spectrum Siemens Hicom 300E/300H Siemens Hipath 4000 CSTA 3 SIP Communication Server Tenovis Integral 33 Substitute	1 DN: • 1 Extension *or* • 1 ACD Position	No constraint	1 Voice DN (Extension number or ACD Position number)
Alcatel-Specific			
Alcatel A4400 standard	In switch: • 1 Extension • 1 ACD Position In place: • Shortcut to Extension • Shortcut to ACD Position	Login ID equal to ACD Position number	1 Voice DN (ACD Position number) `agent substitute= false`

Table 3: Switch-Specific Support (Continued)

Switches	DN in Configuration Manager	Agent Login in Configuration Manager	DN ID Reflected
Alcatel A4400 Agent Substitute	In switch: • 1 Extension • 1 ACD Position In place: • Shortcut to Extension only	Login ID equal to ACD Position number	(T-server option: `agent-substitute=true`) Extension if logged out Position if logged in DNs are removed or added accordingly
Alcatel A4400 Agent emulated	In switch: • 1 Extension In place: • Shortcut to Extension	No defined position for login ID	1 Voice DN (Extension) `agent-substitute=true/false`

Appendix

C Locating your SDK Component's Files

This appendix includes information on where to locate the compressed installation files, readme file, and templates on the installation CD-ROM for the Agent, Queued, and Media Interaction SDK components. The chapter contains the following sections:

Locating the AIL Component's Files

The required files for the AIL SDK can be found in the following location on the CD-ROM:

- `documentation/`, where you can find the readme file.
- `AgentInteraction\Java\windows\`, where Windows setup is located.
- `AgentInteraction/Java/aix/`, where AIX setup is located.
- `AgentInteraction/Java/hp-ux/`, where HP-UX setup is located.
- `AgentInteraction/Java/linux/`, where Linux setup is located.
- `AgentInteraction/Java/solaris/`, where Solaris setup is located.
- `AgentInteraction/Java/tru64UNIX/`, where Tru64 setup is located.
- `templates/`, where template files are located.

Locating the MIL Component's Files

The required files for the MIL SDK can be found in the following location on the CD-ROM:

- `documentation/`, where you can find readme files.
- `MediaInteraction\Java\windows\`, where Windows setup is located.
- `MediaInteraction/Java/aix/`, where AIX setup is located.
- `MediaInteraction/Java/hp-ux/`, where HP-UX setup is located.
- `MediaInteraction/Java/linux/`, where Linux setup is located.
- `MediaInteraction/Java/solaris/`, where Solaris setup is located.
- `MediaInteraction/Java/tru64UNIX/`, where Tru64 setup is located.
- `templates/`, where template files are located.

Locating the QIL Component's Files

The required files for the QIL SDK can be found in the following location on the CD-ROM:

- `documentation/`, where you can find readme files.
- `QueuedInteraction\Java\windows\`, where Windows setup is located.
- `QueuedInteraction/Java/aix/`, where AIX setup is located.
- `QueuedInteraction/Java/hp-ux/`, where HP-UX setup is located.
- `QueuedInteraction/Java/linux/`, where Linux setup is located.
- `QueuedInteraction/Java/solaris/`, where Solaris setup is located.
- `QueuedInteraction/Java/tru64UNIX/`, where Tru64 setup is located.
- `templates/`, where template files are located.

Appendix

D Configuring the Options Tab

This appendix includes information on configuring options for Agent, Queued, and Media Interaction SDK components. The chapter contains the following sections:

Configuring the Options Tab for AIL

In Configuration Manager, configuration options are stored within *section* folders. The section folders and options are stored on the `Options` tab; you click a section to bring up its options. Options are described in detail below.

log Section

Miscellaneous Traces.

console

Default Value: `info`
Valid Values: `false, debug, info, warn, error, fatal`
Changes Take Effect: Immediately.

Level and size of traces to display on the standard output.

file

Default Value: `info, ail, 10MB, 20, zip`
Valid Values: *<level>*, *<file_name_root>*, *<file_max_size>*, *<file_number>* `[, zip] [, timestamped]`

- <*level*>: `false, debug, info, warn, error, fatal`
- <*file_name_root*>: correct path to a file name
- < *file_max_size*>: maximum file size in MB
- < *file_ number*>: number of files for the rolling logs
- `[, zip]`: to get compressed log files (optional)
- `[, timestamped]`: to add a timestamp when a new file is created. An existing file will be replaced when it reaches *file_max_size* (optional)

Changes Take Effect: Immediately.

Specifies how to write entries in log files.

filter

Default Value: `info, 5000`
Valid Values Syntax Format: <*level*>,<*number*>

- <level>: `false, debug, info, warn, error, fatal`
- <number>: any positive integer from `200-10000`

Changes Take Effect: Immediately.

Level of traces to be buffered for internal purposes.

log-filter Section

For Filtering Sensitive Data to Logs.

default-filter-type

Default Value: `copy`
Valid Values: `copy, hide, skip`
Changes Take Effect: Immediately.

Specifies the default way of logging key/value information:

`copy`—copy each pair's key and value to the log.

`hide`—copy keys to the log, but replace values with strings of asterisks.

`skip`—do not copy key/value pairs to the log.

log-filter-data Section

For Filtering Sensitive Data to Logs

<key-name>

Default Value: `copy`
Valid Values: `copy, hide, skip`
Changes Take Effect: Immediately.

Specifies how to log key/value pairs for the specified <*key-name*> key:

`copy`—copy key/value pairs to the log.

`hide`—copy the key to the log, but replace values with strings of asterisks.

`skip`—do not copy key/value pairs to the log for the specified key.

license Section

attempts-interval

Default Value: `5` (in seconds)
Valid Values: <*Any positive integer*>
Changes Take Effect: Immediately.

Time interval, in seconds, between two successive connection attempts.

attempts-max

Default Value:`10`
Valid Value: <*Any positive integer*>
Changes Take Effect: Immediately.

Maximum number of successive connection attempts to the server before triggering an exception.

license-file

Default Value: `license.dat`
Valid Values for Windows:
`<licenseserver_port1@hostname1>;<licenseserver_port2@hostname2>`
Changes Take Effect: Immediately.

A semi-colon-separated list of addresses of Flexlm license servers or explicit paths to the actual license file.
Valid Values for UNIX:
`<licenseserver_port1@hostname1>:<licenseserver_port2@hostname2>`

A colon-separated list of addresses of Flexlm license servers or explicit paths to the actual license file.

dn-at-switch Section

enabled

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After restart.

Used when there are several switches declared in the same configuration. For example, the switch DN ID becomes `103@Xswitch`.

If this option is not present, the default value is set to `true` to ensure 7.5 voice compatibility.

loading Section

This section is only available for the Third Party Application, not for the Third Party Server.

on-demand

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After restart.

Specifies the method of loading configuration objects (optimize client application mode):

`true`—load the configuration objects when needed.

`false`—load all the configuration for agent, DN, and place.

`Note:` Advised for client template.

---

srl-on-demand

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After restart.

Specifies the method to load SRL objects (optimize application mode).

If set to `true`: SRL trees are loaded on demand.

If set to `false`: SRL trees are loaded on startup.

`Note:` Server template.

---

multimedia Section

chat-addp-protocol

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately.

A value of `true` means use the `addp` protocol to connect to the chat server.

chat-busy-threshold

Default Value: `1`
Valid Value: <*Any integer*>
Changes Take Effect: Immediately.

Threshold number of chat interactions after which an agent is seen as busy.

`Note:` Not connected to routing strategy.

---

collaboration-workbin

Default Value: `desktop-collaboration-email-workbin`
Valid Values: <*Any string*>
Changes Take Effect: Immediately.

Used for desktop collaboration feature. When inviting an agent in `pull` mode, the internal invitation is stored in the agent workbin.

email-address-rfc822-strict

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately.

AIL checks if the e-mail addresses of an interaction are compliant with the RFC-822 Standard for the format of ARPA Internet text messages.

email-busy-threshold

Default Value: `1`
Valid Values: <*Any integer*>
Changes Take Effect: Immediately.

Threshold number of e-mail interactions after which an agent is seen as busy.

`Note:` Not connected to routing strategy.

---

email-quote-char

Default Value: `>`
Valid Values: <*Any string*>
Changes Take Effect: Immediately.

This string precedes each line of the sender's message when the agent replies to the e-mail.

email-quote-header

Default Value: `On` *`<date>`*`,` *`<contact>`* `wrote:`
Valid Values: <*Any string*>
Changes Take Effect: Immediately.

Specifies the header that precedes the sender's message. The header can contain two dynamic values, which are *`<date>`* and *`<contact>`*.

enable-multicharset-environment

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After restart.

If set to `true`, unicode data will be retrieved from Contact Server. If set to `false`, non-unicode data will be retrieved from Interaction Server.

open-media-saved-list

Default Value: `none`
Valid Values: Media types separated by commas
Changes Take Effect: Immediately.

A comma-separated list of valid media types that will be saved in UCS. Valid media types are created and configured in the Business directory of Configuration Manager.

reconnect-delay

Default Value: `5`
Valid Values: An integer specifying the reconnection delay in seconds
Changes Take Effect: Immediately.

If the option is not set to `0`, and if AIL switches to the backup Interaction Server, it waits the reconnection delay before attempting to connect.

logout-on-last-media

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately.

When set to true, the place is automatically logged out of the Interaction Server when the last media is logged out. Otherwise, it is possible to remain logged-in from an Interaction Server point of view, without any media.

network Section

alternate-locations

Default Value: ""
Valid Values: Switch names separated by commas
Changes Take Effect: Immediately.

A comma-separated list of switch locations for which network alternate calls are enabled.

auto-reconnect-timeout

Default Value: `0`
Valid Values: An integer specifying the timeout in seconds, `0` to deactivate
Changes Take Effect: Immediately.

If non-zero, a failed network consult is automatically reconnected after the specified timeout in seconds.

conference-locations

Default Value: ""
Valid Values: Switch names separated by commas
Changes Take Effect: Immediately.

A comma-separated list of switch locations for which network complete conference is enabled.

consult-location

Default Value: ""
Valid Values: Switch names separated by commas
Changes Take Effect: Immediately.

A comma-separated list of switch locations for which network consultation call is enabled.

enable-synchronized-reconnect

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately.

If `true`, network reconnect is possible only after receiving a network response to a network consult.

reconnect-locations

Default Value: ""
Valid Values: Switch names separated by commas
Changes Take Effect: Immediately.

A comma-separated list of switch locations for which network reconnect feature is enabled.

reroute-locations

Default Value: ""
Valid Values: Switch names separated by commas
Changes Take Effect: Immediately.

A comma-separated list of switch locations for which the reroute feature is enabled.

single-step-transfer-locations

Default Value: ""
Valid Values: Switch names separated by commas
Changes Take Effect: Immediately.

A comma-separated list of switch locations for which network single step transfer feature is enabled.

transfer-locations

Default Value: ""
Valid Values: Switch names separated by commas
Changes Take Effect: Immediately.

A comma-separated list of switch locations for which network complete transfer is enabled.

outbound Section

enable-chain-75api

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: After restart.

If set to `true`, enables the use of the new 7.5 API with the `OutboundChain` class, and abandons the `InteractionVoiceOutbound` class.

signature Section

include-agent-name

Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: Immediately.

If `true`, add the agent name at the begin of the signature of an e-mail, after the prefix.

line1, line2....lineN

Default Value: ""
Valid Values: <*Any string*>
Changes Take Effect: Immediately.

Used to define the lines of an e-mail signature.

prefix

Default Value: "--"
Valid Values: <*Any string*>, <*empty string*>
Changes Take Effect: Immediately.

Use this option to set a separator string before your signature. If this option is set to <*empty string*>, there is no additional line in the standard signature.

voice Section

a4400-custom-substitute-mode

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After restart.

Use this option to customize substitute behavior. When set to true, the virtual position DN that is created when an agent logs in will be replaced with the extension DN in the place.

database

Default Value: `all`

Valid Values: `all, external, manual, none`
Changes Take Effect: Immediately.

This option specifies the use of the UCS database for voice calls when Contact Server DB is connected.

`all`—Any voice call uses the database.

`external`—Internal calls do not use the database.

`none`—No voice call uses the database.

`manual`—AIL will no longer manage the automatic contact lookup and the creation of voice interactions in the UCS database. In this mode, your application is responsible for the following actions:

- Fetching (or creating) the `ContactServerId DBID` for each contact by using another Genesys component, or by calling the `contactManager.findOrCreateContact()` AIL method.
- Attaching the `ContactServerId DBID` to the AIL voice interaction by calling the `InteractionVoice.setContactId()` method.
- Saving contact records and interactions in the UCS database by calling the `InteractionVoice.save()` or `InteractionVoice.markDone()` AIL methods.

dms-last-digits

Default Value: `-1`
Valid Values: <*Any positive integer*>
Changes Take Effect: Immediately.

For Nortel Communication Server 2000 (DMS 100) switch only. Specifies how many digits should be kept at the end of a DN number to get its dialable number. For example, if the DN number is 1001234567 and this option is set to `4`, this DN is called from 4567. If the value is `-1` or if the resulting transformation does not provide a correct number, the former number is used.

enable-all-routing-events

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately.

Option to send all events to `RoutingInteractionlisteners`. Default is `false` (to receive only `NEW`, `IDLE`, and `INFO-CHANGES` events). Set to `true` to receive all the events.

enable-attached-data-for-transfer

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately.

Use this option to disable the attachment of `GCS_*` data when transferring a phone call.

enable-interaction-id-tracking

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately.

Enables every means necessary to maintain uniqueness of interaction IDs.

enable-possible-changed-event

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately.

If set to `true`, an interaction sends an `InteractionEvent` with the `POSSIBLE_CHANGED` event reason, if the interaction's possible actions have changed due to the status change of another related interaction (consult, primary). If set to `false`, you will not receive those events.

idle-interactions-cleanup-delay

Default Value: `-1`
Valid Values: *<Any integer>*
Changes Take Effect: Immediately.

The length of time, in minutes, that an interaction can be `IDLE` before it can be silently removed when one or more further interactions become `IDLE`. This option is intended to prevent memory leaks by removing `IDLE` interactions in AIL. Set to `-1` to disable this option.

complete-on-held-conn-id

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When a request for complete transfer or complete conference is sent to TServer, two connection IDs must be passed in two parameters. AIL passes the connection ID of the main call as the first argument, and that of the consult call as the second argument.

If set to `true`, this option reverts the previous behavior of AIL, which was to pass the connection ID of the call that is currently on hold as the first argument and that of the call currently talking as the second argument.

kworker Section

auto-markdone

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately.

If set to `true`, an interaction on the CTI-Less T-Server is automatically marked `done` when it is released.

easy-newcall

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately.

If set to `true`, a newly created interaction is automatically dialed.

reroute

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately.

Specifies whether the reroute capability is enabled or disabled.

reroute-location

Default Value: `<location>`
Valid Values: *switch1,...,switchN*
Changes Take Effect: Immediately.

Specifies which switch locations can receive rerouted calls. This option is used only if the `reroute` option is set to `true`. If set to the default value, rerouting is to any switch. If set to another valid value, rerouting is restricted to the specified switches.

Configuring the Options Tab for MIL

In Configuration Manager, configuration options are stored within *section* folders. The section folders and options are stored on the `Options` tab; you click a section to bring up its options. Options are described in detail below.

log Section

omsdk-console

Default Value: `info`
Valid Values: `false, debug, info, warn, error, fatal`
Changes Take Effect: After restart.

Level and size of traces to display on the standard output.

Non-mandatory

omsdk-file

Default Value: `info, omsdk.log, 10, 20`
Valid Values: `<level>, <filename>, <max filesize>, <max filenumber>`

- <level>: false, debug, info, warn, error, fatal
- <file name>: correct path to a file name
- <max file size>: maximum file size in MB
- <max file number>: number of files for the rolling logs

Changes Take Effect: After restart.

Used to put traces of OMSDK in file. If this option is not specified, OMSDK logging to the file is not used.

Non-mandatory

omsdk-msgsrv

Default Value: off
Valid Values: off, debug, info, warn, error, fatal
Changes Take Effect: After restart.

Specifies the level and size of OMSDK to display in the MessageServer centralized log. If this option is not specified, OMSDK logging to MessageServer is not used.

Non-mandatory

log-filter Section

For Filtering Sensitive Data to Logs

default-filter-type

Default Value: copy
Valid Values: copy, hide, skip
Changes Take Effect: After restart.

Specifies the default way of logging key/value information:

copy—copy each pair's key and value to the log.

hide—copy keys to the log, but replace values with strings of asterisks.

skip—do not copy key/value pairs to the log.

log-filter-data Section

For Filtering Sensitive Data to Logs

<key-name>

Default Value: copy
Valid Values: copy, hide, skip
Changes Take Effect: After restart.

Specifies how to log key/value pairs for the specified <*key-name*> key:

copy—copy key/value pairs to the log.

`hide`—copy the key to the log, but replace values with strings of asterisks.

`skip`—do not copy key/value pairs to the log for the specified key.

esp Section

esp-response-timeout

Default Value: 2000
Valid Values: <*Any positive integer*>
Changes Take Effect: After restart.

Used to specify response timeout for ESP service. If client's ESP handler does not manage to handle a request during a specified period, MIL sends a fault message as a response. Specified in milliseconds.

settings Section

enable-attached-data-byte-array

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After restart.

Specifies how to perform `TKVList` to `Map` conversion (for attached data), namely how to convert `KVTypeBinary` values. The possible options are `ArrayList` of `Byte*byte[]`. By default, the option is `true` and `KVTypeBinary` is converted into `ArrayList` of `Byte`.

This option specifies the conversion method for binary data in ESP requests and responses.

If set to `true`, MIL converts binary data from a request to an array of bytes (`byte[]`) and expects that in responses, binary data will be of the same type.

If set to `false`, MIL converts binary data to an `ArrayList` of objects with class `Byte`. Same for responses.

use-esp

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After restart.

Set to `true`, specifies that you use low-level ESP service in MIL.

Non-mandatory

use-lca

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After restart.

Set to `true`, specifies that you use low-level LCA service in MIL.

Non-mandatory

workflow Section

request-timeout

Default Value: `5000`
Valid Values: <*Any positive integer*>
Changes Take Effect: After restart.

Used for specifying a request timeout for the Interaction Server service. If an answer is not received from `IxnServer` within the specified period, a corresponding `TimeoutException` will be thrown. Specified in milliseconds.

Non-mandatory

ucs Section

ucs-identify-create-contact

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After restart.

This option is used in `MILUCSManager.saveInteraction(MILInteraction interaction)` class. When saving interaction into UCS, this parameter shows whether to identify contact and create new one if it is not found.

```
If (IdentifyCreateContact==true)
{- Identify contact and create new contact if not found;- Interaction.ContactId
= found or created Contact.Id}
else {Interaction.ContactId = UserData.ContactId}
```

Non-mandatory

Configuring the Options Tab for QIL

In Configuration Manager, configuration options are stored within *section* folders. The section folders and options are stored on the `Options` tab; you click a section to bring up its options. Options are described in detail below.

general Section

agent-place

Default Value: `""`
Valid Values: A string representing an existing system`Place` (a `Place` that is not used by an `Agent`).
Changes Take Effect: After restart.

Used to connect to Interaction Server, to make it possible to perform ad-hoc features.

log Section

omsdk-console

Default Value: `info`
Valid Values: `false, debug, info, warn, error, fatal`
Changes Take Effect: After restart.

Level and size of traces to display on the standard output.

Non-mandatory.

omsdk-file

Default Value: `info, omsdk.log, 10, 20`
Valid Values: `<level>, <filename>, <max filesize>, <max filenumber>`

- `<level>`: `false, debug, info, warn, error, fatal`
- `<file name>`: correct path to a file name
- `<max file size>`: maximum file size in MB
- `<max file number>`: number of files for the rolling logs

Changes Take Effect: After restart.

Used to put OMSDK trace messages in the file. If this option is not specified, OMSDK logging to the file is not used.

Non-mandatory.

omsdk-msgsrv

Default Value: `off`
Valid Values: `off, debug, info, warn, error, fatal`
Changes Take Effect: After restart.

Specifies the level and size of OMSDK to display in the MessageServer centralized log. If this option is not specified, OMSDK logging to MessagerServer is not used.

Non-mandatory.

log-filter Section

For Filtering Sensitive Data to Logs

default-filter-type

Default Value: `copy`
Valid Values: `copy, hide, skip`
Changes Take Effect: After restart.

Specifies the default way of logging key/value information:

`copy`—copy each pair's key and value to the log.

`hide`—copy keys to the log, but replace values with strings of asterisks.

`skip`—do not copy key/value pairs to the log.

log-filter-data Section

For Filtering Sensitive Data to Logs

<key-name>

Default Value: `copy`
Valid Values: `copy, hide, skip`
Changes Take Effect: After restart.

Specifies how to log key/value pairs for the specified <*key-name*> key:

`copy`—copy key/value pairs to the log.

`hide`—copy the key to the log, but replace values with strings of asterisks.

`skip`—do not copy key/value pairs to the log for the specified key.

settings Section

enable-attached-data-byte-array

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After restart.

This option specifies the conversion method for binary data in ESP requests and responses.

If set to `true`, QIL converts binary data from a request to an array of bytes (`byte[]`) and expects that in responses, binary data will be of the same type.

If set to `false`, QIL converts binary data to an `ArrayList` of objects with class `Byte`. Same for responses.

workflow Section

request-timeout

Default Value: `5000`
Valid Values: <*Any positive integer*>
Changes Take Effect: After restart.

Used for specifying a request timeout for the Interaction Server service. If an answer is not received from `IxnServer` within the specified period, a corresponding `TimeoutException` will be thrown. Specified in milliseconds.

Non-mandatory.

Index

K

L

M

N

O

P

R

S

T

U

V

W