GENESYS™

AIL Deployment Guide

Genesys Interaction SDK 7.6.6

1/13/2022

Table of Contents

Interaction SDK Java Deployment Guide

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).

Warning
This document is valid only for the **7.6.6** release(s) of this product.

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.

Important
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

Chapter Summaries

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

- AIL, MIL, and QIL Basics, provides information about which systems you must set up before AIL, MIL, and QIL installation and configuration.

- Installing and Configuring Interaction SDK Components indicates how to install AIL, MIL, and QIL via the Genesys Interaction SDK Installation Package CD.

- Custom Applications, gives details about installing and configuring your custom built application to run in a Genesys environment.

- Install and Configure Custom Applications

- Switch-Specific Support Configuration, provides the information for switch-specific configuration.

- Locating your SDK Component Files, gives details on where to locate your required files for your SDK component.

- Configuring the Options Tab, provides the information needed to configure the options tab of each Genesys Interaction SDK component.

List of Procedures

- Preparing to Configure your Interaction SDK Component

- Importing an Application Template into Configuration Manager
- Configuring your Interaction SDK for Deployment
- Launching System Installation
- Installing your Interaction SDK Component
- Setting up VoIP Support
- Setting up your SIP Communication Server
- Configuring for your Specific Switch
- Running MIL and QIL on the Same JVM
- Starting MIL in Server Mode
- Configuring External Service Protocol Request
- Configuring the Tenants Tab for N-Tier Deployment of AIL using the Standard Method
- Configuring the Tenants Tab for N-Tier Deployment of AIL using Alternative Methods
- Configuring Contact Look Up or Creating Voice Interactions

Table of Contents

See the Change History page to get the list of modifications for this document.

Change History

This section lists all the changes between the 7.6.5 and 7.6.6 versions of this document.

Version 7.6.609.00

Page name	State	Additional details
Welcome	Updated	Section The Interaction SDK Set was updated.

AIL, MIL, and QILBasics	Updated	• New sections
		  • New in this Release
		  • Release 7.6.609.00
		• Updated sections
		  • API Roles and Components

Installing and Configuring	Updated	• Updated sections
		  • Preparing to Configure your Interaction SDK Component
		  • Importing an Application Template into Configuration Manager
		  • Configuring your Interaction SDK for Deployment
		  • Windows Installation

Custom Applications	Updated	• New sections: • Configuring TLS
The Properties Tab	Updated	Section `Connections Tab` was updated
Configuring the Options Tab For AIL	Updated	• New sections • custom-fields-include-all-fields

Page name	State	Additional details
		• enable-auto-markdone-on-transfer • commons-connection • string-attributes-encoding • settings • Updated sections • reroute-location

AIL, MIL, and QIL Basics

This page 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.

API Roles and Components

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.)

Agent Interaction (Java API)

API Role

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.

Components

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.
- A set of code examples that exercise some important features of the API, delivered in `.zip` and `.tar.gz` format.

Media Interaction (Java API)

API Role

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.

Important

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

MIL Components

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.

- A set of code examples that exercise some important features of the API, delivered in `.zip` and `.tar.gz` format.

Queued Interaction (Java API)

API Role

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.

QIL Components

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.
- A set of code examples that exercise some important features of the API, delivered in `.zip` and `.tar.gz` format.

New in this Release

Release 7.6.609.00

- AIL now supports Red Hat 6.0 32-bit or 64-bits, Windows Server 2012 64-bits, and Windows 8 32-bits or 64-bits.
- AIL no longer supports RedHat Linux 4, Windows 2000, and Tru64 O/S.
- AIL now supports several character encodings for connections to Genesys Servers. AIL exposes this feature in the new commons-connection/string-attributes-encoding which can be set to the appropriate encoding value. Note that this option does not affect the connection to the Configuration Server. To modify the character encoding for the Configuration Server's connection, use the -Dpsdk.connection.charset JVM option.
- AIL now uses Java Generics, and AIL methods return typed Collections and Maps. Note: This feature does not require you to modify applications compiled with a former version of AIL.
- AIL can now add, remove, and change an agent's level of skills. AIL exposes this feature in the following new methods.
    - In the AilFactory class:

```
public Collection<Skill> getSkills();
public Skill getSkill(Integer skillId);
```

- • In the Agent class:

```
public Collection<Skill> getSkills();
public void addSkill(int skillId, int level);
public void setSkillLevel(int skillId, int skillLevel);
public void removeSkill(int skillId);
```

- AIL now allows agents to force the multimedia logout of other agents. AIL exposes this feature in the
  Place.forceLogout(String targetEmployeeId, String targetPlace); method.

- AIL now supports the modification of the JVM default XML parser.

- AIL now provides TLS for connections to Genesys backend servers.

- AIL now allows you to use Contact PSDK rather than UCS API to handle the connection and requests to
  UCS.

Tasks and Their Related Procedures

The following table summarizes the tasks and their related procedures addressed in this guide.

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 Configuring your Interaction SDK for Deployment Configuring the Properties Tab Configuring the Options Tab for AIL
To install your AIL Interaction SDK component for use in developing custom applications.	Launching System Installation Installing your Interaction SDK Component
To set up your environment so that you can develop using your QIL Interaction SDK component.	Preparing to Configure your Interaction SDK Component Configuring your Interaction SDK for Deployment Configuring the Properties Tab Configuring the Options Tab for QIL
To install your QIL Interaction SDK component for use in developing custom applications.	Launching System Installation Installing your Interaction SDK Component
To set up your environment so that you can develop using your MIL Interaction SDK component.	Preparing to Configure your Interaction SDK Component Configuring your Interaction SDK for Deployment Configuring the Properties Tab Configuring the Options Tab for MIL
To install your MIL Interaction SDK component for use in	Launching System Installation

Objective	Related Procedures and Actions
developing custom applications.	Installing your Interaction SDK Component
To allow your custom application to run in a Genesys environment.	Configuring TLS Setting up VoIP Support Setting up your SIP Communication Server Configuring for your Specific Switch Running MIL and QIL on the Same JVM Starting MIL in Server Mode Configuring External Service Protocol Request

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.

> ### Important
> 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.

template names should be changed

Configuration and Installation

1. Preparing to Configure your Interaction SDK Component

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.
Start

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_766.apd`
          `Agent_Interaction_server_766.apd`

        - MIL:
          `Media_Interaction_SDK_server_766.apd`

        - QIL:
          `Queued_Interaction_server_766.apd`

    End
    Next Steps

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

2. Importing an Application Template into Configuration Manager (Optional)

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 7.6 Java Developer's Guide for configuration environment alternatives.

Start

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_766.apd`
      `templates/Agent_Interaction_server_766.apd`

    - MIL:
      `templates/Media_Interaction_server_766.apd`

    - QIL:
      `templates/Queued_Interaction_server_766.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
Next Steps

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

3. Configuring your Interaction SDK for Deployment

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.

Start

1. In Configuration Manager, open the Environment folder and select the Applications folder.

2. From the File menu, selectNew > Application. The Browse window opens, listing all application templates present in Configuration Manager.

3. Choose the application template.

    - AIL:
      Agent_Interaction_server_766.apd
      Agent_Interaction_client_766.apd.

    - MIL:
      Media_Interaction_SDK_server_766.apd

    - QIL:
      Queued_Interaction_server_766.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 The Properties Tab.

End
Next Steps

- You will need to configure the properties tab. Follow the guidelines in The Properties Tab.

- You will need to configure the options tab. Follow the guidelines in Configuring the Options Tab.

3. Launching System Installation

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.

- 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: Locating the AIL Component Files.

  - MIL: Locating the MIL Component Files.

  - QIL: Locating the QIL Component Files.

  Windows Installation

  Start

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 7.6.6\AIL\
   - MIL:
     Program Files\GCTI\Interaction SDK for Java 7.6.6\MIL\
   - QIL:
     Program Files\GCTI\Interaction SDK for Java 7.6.6\QIL\

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

End
Next Steps

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

UNIX Installation

Prerequisites

- Check to make sure you have installation rights.

Start

1. Navigate to the component's directory on the CD.

    - AIL:
      `AgentInteraction/Java/<os>/`

2. MIL:
   `MediaInteraction/Java/<os>/`

3. QIL:
   `QueuedInteraction/Java/<os>/`

4. Run the `install.sh` script.

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

> ### Important
> 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
Next Steps

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

4. Installing your Interaction SDK Component

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.

Start

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. The following table lists the libraries added during system installation.

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
Next Steps

- You can now use your Interaction SDK component to create custom built applications. For examples, see Custom Applications.

Custom Applications

This chapter describes how to install and configure your custom built applications to run in a Genesys environment. 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.

Configuring TLS

In 7.6.6, AIL integrates the PSDK Application Template Application Block, which provides TLS support. Before you can use and test your AIL application, you must configure secured ports as detailed in the Deployment Guide.

For further details, you should read Platform SDK Implementation of TLS chapter which give all details about TLS in Platform SDK.

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 Installing and Configuring Interaction SDK Components.

- 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

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` .

> **Important**
>
> 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
Next Steps

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

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 Installing and Configuring Interaction SDK Components.

Start

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.

   > **Important**
   >
   > 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.

> ### Important
> The realm for password authentication is configured globally. There is one realm per Communication Server.

End
Next Steps

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

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 Installing and Configuring Interaction SDK Components.

Start

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, Switch-Specific Support Configuration for details.

> ### Important
>
> 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
Next Steps

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

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 Installing and Configuring Interaction SDK Components.

Start

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.

> ### Important
>
> Genesys advises you to use the MIL template and complete it with the applicable
> QIL options.

End
Next Steps

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

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 Installing and Configuring Interaction SDK Components.

Start

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`

e. `commons-lang-2.0.jar`

f. `concurrent.jar`

g. `log4j-1.2.8.jar`

h. `mail.jar`

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

End
Next Steps

- Customize your call center further by:

   - Setting up VoIP Support.

   - Setting up your SIP Communication Server.

   - Configuring for your Specific Switch.

   - Running MIL and QIL on the Same JVM.

   - Configuring External Service Protocol Request.

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 Installing and Configuring Interaction SDK Components.
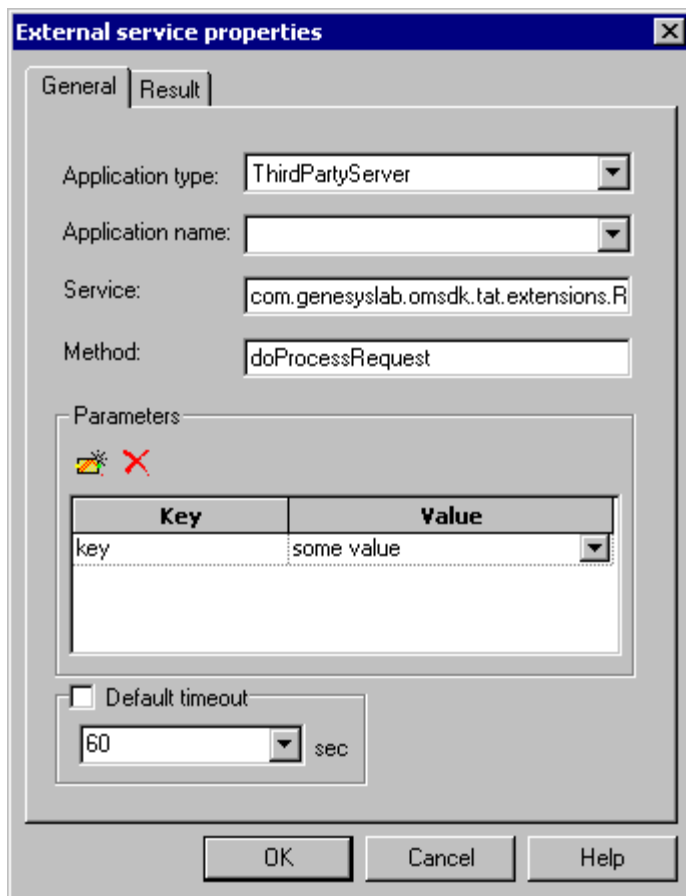
Start

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 External Service Properties Window 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.

> ## Important
> You can create any key/value pair.

External Service Properties Window

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

> **Important**
> You can choose the default value.

7. Click OK to save your data.

> **Important**
> 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
Next Steps

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

The Properties Tab

This appendix includes information on how to configure the properties tab of for Agent, Queued, and Media Interaction SDK components.

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

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

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)

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

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.

> ### Important
> Since 7.6.4, AIL supports only one Stat Server connection

- 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

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: Configuring the Options Tab for AIL as a guide to viewing or changing AIL options.
- MIL: Configuring the Options Tab for MIL as a guide to viewing or changing MIL options.
- QIL: Configuring the Options Tab for QIL 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

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

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.

> ### Important
> 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.)

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.

Prerequisites

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

Start
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
   ```

   > **Important**
   >
   > 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
Next Steps

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

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 Installing and Configuring Interaction SDK Components.

Start

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:

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

   To associate the Person's account with the Application:

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

End
Next Steps

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

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 Installing and Configuring Interaction SDK Components.

Start

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`

    > **Important**
    >
    > 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` created in the voice-attribute section of the AIL application must have the same name than 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.

3. To customize the contact attributes, or to create new ones, for voice interactions in Configuration Manager and in the appropriate tenant, configure the `voice-attribute` option for Voice interactions:

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

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

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

voice-attribute Section

BusinessID

- AccountNb
- **Description:** 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.

> ## Important
> 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
Next Steps

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

Switch-Specific Support Configuration

For full support of the following switches, configure the `Place` that the agent will log into as described in Switch-Specific Support.

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)
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	1 DN: • 1 Extension *or* • 1 ACD Position	No constraint	1 Voice DN (Extension number or ACD Position number)

Switches	DN in Configuration Manager	Agent Login in Configuration Manager	DN ID Reflected
Tenovis Integral 33 Substitute			
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`
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`

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.

Locating the AIL Component 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.
- `templates/`, where template files are located.

Locating the MIL Component 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.
- `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.

- `templates/`, where template files are located.

Configuring the Options Tab

This appendix includes information on configuring options for Agent, Queued, and Media Interaction SDK components.

- Configuring the Options Tab for AIL
- Configuring the Options Tab for MIL
- Configuring the Options Tab for QIL

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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** Time interval, in seconds, between two successive connection attempts.

attempts-max

- Default Value:10

- Valid Value: *<Any positive integer>*

- Changes Take Effect: Immediately.

- **Description:** Maximum number of successive connection attempts to the server before triggering an exception.

license-file

- Default Value: `license.dat`
- Changes Take Effect: Immediately.

- Valid Values for Windows:
  - <licenseserver_port1@hostname1>;<licenseserver_port2@hostname2>
  - **Description:** 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>
  - **Description:** 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.

- **Description:** 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.

- **Description:** 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.

> ### Important
> Advised for client template.

srl-on-demand

- Default Value: `true`

- Valid Values: `true, false`

- Changes Take Effect: After restart.

- **Description:** 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.

> ### Important
> Server template.

multimedia Section

chat-addp-protocol

- Default Value: `false`
- Valid Values: `true, false`
- Changes Take Effect: Immediately.
- **Description:** 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.
- **Description:** Threshold number of chat interactions after which an agent is seen as busy.

> ### Important
> Not connected to routing strategy.

collaboration-workbin

- Default Value: `desktop-collaboration-email-workbin`
- Valid Values: *<Any string>*

- Changes Take Effect: Immediately.

- **Description:** 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.

- **Description:** 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.

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

> ### Important
> Not connected to routing strategy.

email-quote-char

- Default Value: >

- Valid Values: *<Any string>*

- Changes Take Effect: Immediately.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** Define the reconnection delay. 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

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

consult-locations

- Default Value: ""

- Valid Values: Switch names separated by commas

- Changes Take Effect: Immediately.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

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

custom-fields-include-all-fields

- Default Value: `false`

- Valid Values: `true` , `false`

- Changes Take Effect: After restart.

- **Description:** If set to `true`, enables the `InteractionVoiceOutbound.getCustomFields()` methods to return all the fields defined in the custom fields.

signature Section

include-agent-name

- Default Value: `true`

- Valid Values: `true` , `false`

- Changes Take Effect: Immediately.

- **Description:** 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.

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

prefix

- Default Value: `"--"`

- Valid Values: *<Any string>* , *<empty string>*

- Changes Take Effect: Immediately.

- **Description:** 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.

- **Description:** 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`
    - `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.

- Changes Take Effect: Immediately.

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

dms-last-digits

- Default Value: `-1`

- Valid Values: *<Any positive integer>*

- Changes Take Effect: Immediately.

- **Description:** 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.

- **Description:** 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.

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

enable-auto-markdone-on-transfer

- Default Value: `true`

- Valid Values: `true, false`

- Changes Take Effect: Immediately.

- **Description:** Use this option to automatically mark done transferred calls.

enable-interaction-id-tracking

- Default Value: `true`

- Valid Values: `true, false`

- Changes Take Effect: Immediately.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** If set to `true` , a newly created interaction is automatically dialed.

reroute

- Default Value: `false`

- Valid Values: `true, false`

- Changes Take Effect: Immediately.

- **Description:** Specifies whether the reroute capability is enabled or disabled.

reroute-location

- Default Value: `<location>`

- Valid Values: switch1,...,switchN

- Changes Take Effect: Immediately.

- **Description:** 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.

commons-connection

string-attributes-encoding

- Default Value: N/A

- Valid Values: String for a valid charset (US-ASCII, UTF-8, UTF-16BE, UTF-16LE, UTF-16, and so on.)

- Changes Take Effect: Immediately.

- **Description:** Specifies the character encoding for connections to Genesys Servers, except for the Configuration Server. This option must be set for each AIL connection in the Connections tab of the CME application:#Select your connection, right click, then navigate to the Advanced tab.#Create the `commons-connection/string-attributes-encoding` option and set the appropriate encoding value.

> ## Important
> You cannot configure the charset encoding for the connection to the Configuration Server in the Configuration Server itself. To change the charset encoding for the connection to the Configuration Server, use the following JVM option:
>
> `-DFile.encoding=UTF-8 -Dpsdk.connection.charset=windows-1252`

settings

ucs-library

- Default Value: `ucsapi`

- Valid Values: ucsapi, psdk

- Changes Take Effect: Immediately.

- **Description:** Specifies the use of the UCS API or Contact PSDK to manage UCS connections.

> ### Important
> You can use Contact PSDK only with UCS 8.1.100.xx or higher. A lower UCS version might lead to incorrect results.

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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.
- **Description:** 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.
- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.

- **Description:** 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.