

# eServices Deployment Guide

Genesys Engage Digital (eServices) 8.5.2

12/29/2021

# Table of Contents

# eServices Deployment Guide

Welcome to the eServices Deployment Guide. This document describes how to deploy your eServices 8.5 solution.

### Overview

This chapter introduces you to eServices and its architecture.

Functional Components

Architecture

### Deployment Planning

This chapter explains how to plan your deployment of the eServices solution.

Deployment Tasks

Using the Configuration Wizards

### Third Party Application

This chapter explains how to deploy the third-party web components for the eServices solution.

Deploying Java

### Deploying eServices

These topics explain how to deploy eServices.

Deploying on Windows

Deploying on UNIX

Manual Deployment

### Other Information

These topics discusses other information required to run your eServices solution.

Transport Layer Security

**Important**

For the latest version of Interation Server Deployment Guide, see here.

For the latest version of Interaction Server Administration Guide, see here.

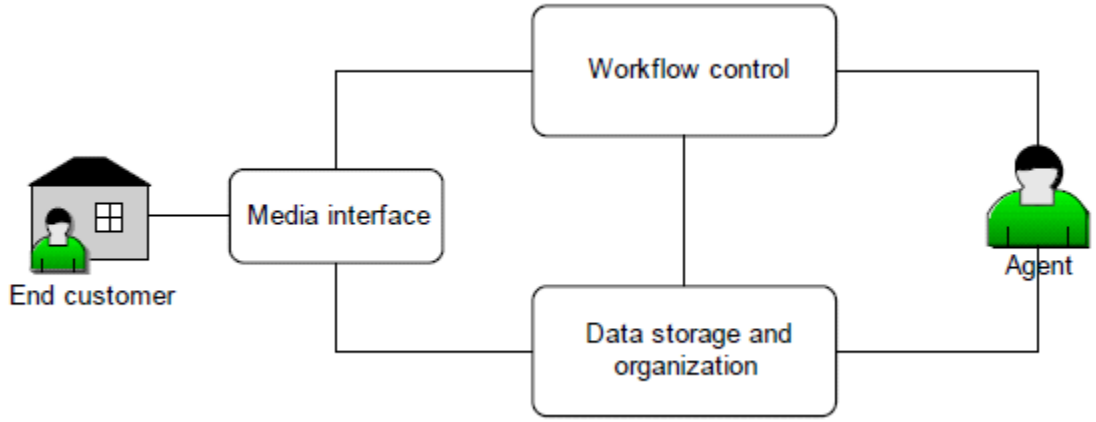
# Overview of eServices

Genesys eServices (called Multimedia in 8.0.0 and earlier) is a platform on which you can assemble a coordinated suite of components that processes, manages, and archives customer/agent interactions in the media of your choice. This chapter introduces

- The components and their functionality
- The basic architecture and the processing of interactions

# Functions and Components

The figure below shows the overall functionality of eServices with any media type. This figure shows functions only, not components.




General Functioning of eServices

The three major functions shown in the figure are:

- The **media interface,** which brings interactions into the system. It may interface with e-mail, chat, or other media.
  **[+]Media Interfaces**

  The media interfaces available with eServices 8.5 are E-mail Server, Chat Server, Social Messaging Server, and SMS Server.

  > **Important**
  > eServices 8.5 also supports the processing of 3rd Party Media interactions with the help of Genesys's 3rd Party Media SDK and Interaction SDK products. See the documentation for those products for more information.

  - E-mail Server interfaces with the enterprise mail server and the Genesys Web API Server, bringing in new e-mail interactions and sending out replies or other outbound messages.
  - Chat Server works with Web API Server to open, conduct, and close chat interactions between agents and customers.
  - SMS Server receives and handles SMS and MMS messages sent from a mobile client. SMS Server uses SMPP v3.4 protocol for SMS support, and MM1, MM7 protocols for MMS support; further details are available.
  - Web API Server works with Interaction Server to create, schedule and close callback requests via the web.
  - Social Messaging Server provides Social Media functionality, such as support for Facebook, Twitter,

and RSS. More information is provided in the Social Media Solution Guide.

To the workflow control components, these interfaces transmit operational data about each interaction, consisting of an identifying code plus some data about the interaction (date received, originating party, and so on).

To the data storage components, they transmit the body of the interaction—that is, a transcript of the e-mail or chat session.

- A **database,** which stores the history of the interaction and associates it with related interactions to form a thread. It also stores contact information and other types of data used at different points in the processing of interactions.
  **[+]Data Storage: Universal Contact Server**

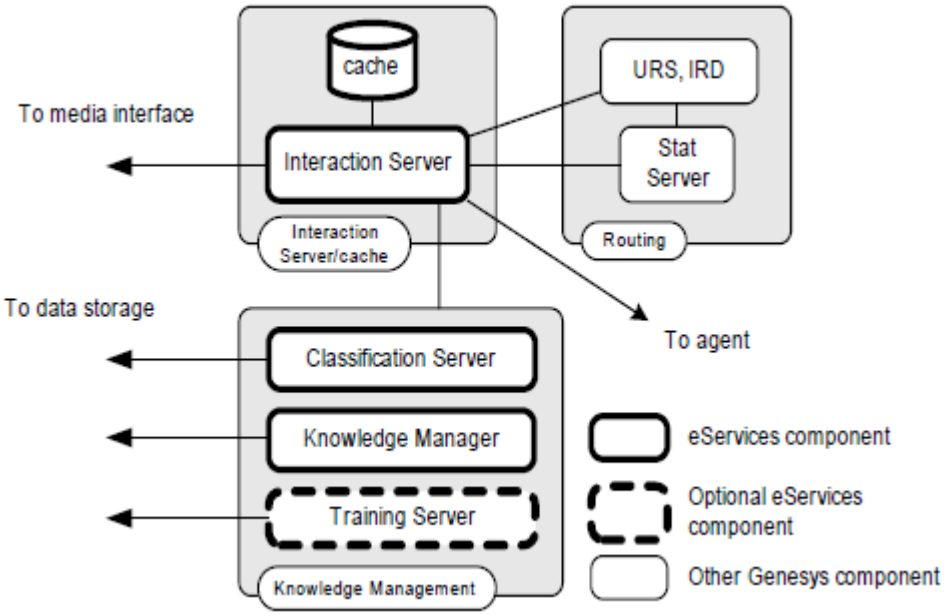
  **Universal Contact Server** (UCS) interfaces with a database that stores the following:

  - Contact information, such as names, addresses, phone numbers
  - Contact history: previous interactions with this contact
  - Other data used in processing interactions, such as standard responses and screening rules.

  Your eServices installation (as part of the Universal Contact Server installation package) includes scripts for setting up the database (Microsoft SQL Server, Oracle, and DB2 are supported. **Universal Contact Server Manager** provides a user interface for setting data-archiving and pruning functions.

- **Workflow control,** which determines where the interaction goes and what happens to it.
  **[+]Workflow Control**

  The components illustrated in the figure below handle workflow control.




Workflow Control Components

Not shown in the figure are other required Genesys Management Framework components, such as Configuration Server and the Management Layer.

The workflow control components fall into three groups, described in the following sections.

## Interaction Server

Interaction Server is the central interchange for interaction flow.

- It receives interaction operational data from the media interface.
- It stores the operational data in a **cache** (a database) while receiving and transmitting information about the interaction. This cache also contains **queues** through which the interaction passes as part of its processing.
- It works in concert with the Routing components to route interactions according to interaction workflows and routing strategies (following section).
- It provides the means for agents to log in and indicate readiness.

> **Important**
>
> For Interaction Server 9.0 documentation, see the Interaction Server product page.

## Routing

Routing components include the following:

- **Interaction Routing Designer (IRD)** and **Universal Routing Server (URS)** design and execute **routing strategies,** which trigger functions such as automatic responses and screening; apply logic (segmentation and conditional branching) to the flow; and ultimately deliver the interaction to an agent or other target. Routing strategies are one of the two main types of objects used in interaction workflows (previous section).
- **Interaction Design,** a subcomponent of IRD, creates and displays **Business Processes,** which plot an overall path for interactions. Interaction workflows map a route for the interaction between contact center objects, principally queues and routing strategies (following section). Interaction workflows are executed by Interaction Server.
- **Stat Server** accumulates data about places, agents, and place/agent groups; converts the data into statistically useful information; and passes these calculations to other software applications. In particular, Stat Server provides information to URS about agents' capacities in terms of the number of interactions, the media type of an interaction, and so on.

## Knowledge Management

**Genesys Knowledge Management** is made up of the following:

- **Classification Server,** which applies screening rules when triggered to do so by a routing strategy. Screening rules are basic pattern-matching queries performed on interaction contents. The results of these queries can then be referred to by further routing strategy logic. In the Genesys Content Analyzer option (see below), Classification Server also applies models to categorize incoming interactions. Both screening rules and models are stored in the Universal Contact Server database.
- **Training Server,** which trains the system to recognize categories. It is active only in the Content Analyzer option (see below).
- **Knowledge Manager,** which is the user interface component for Knowledge Management. You use Knowledge Manager to:
  - Manage the Standard Response Library, which is a collection of ready-made responses to common inquiries and topics.
  - Manage screening rules.
  - Manage categories, which are used to organize standard responses.

**Genesys Content Analyzer** is an optional enhancement to Knowledge Management, requiring a separate license. It uses natural language processing technology to analyze incoming interactions for assignment to the categories of the standard response category system. The statistical tools that enable this analysis, called models, are built up and refined by Training Server as it processes collections of preclassified interactions. Setting up and scheduling these training sessions is another function of Knowledge Manager.

**FAQ** works with Genesys Content Analyzer to convert your category structure and standard responses into an Frequently Asked Questions (FAQ) list. You can either post the resulting list as text on your web site or use it as the source for an automatic question-answering facility.

## Summary

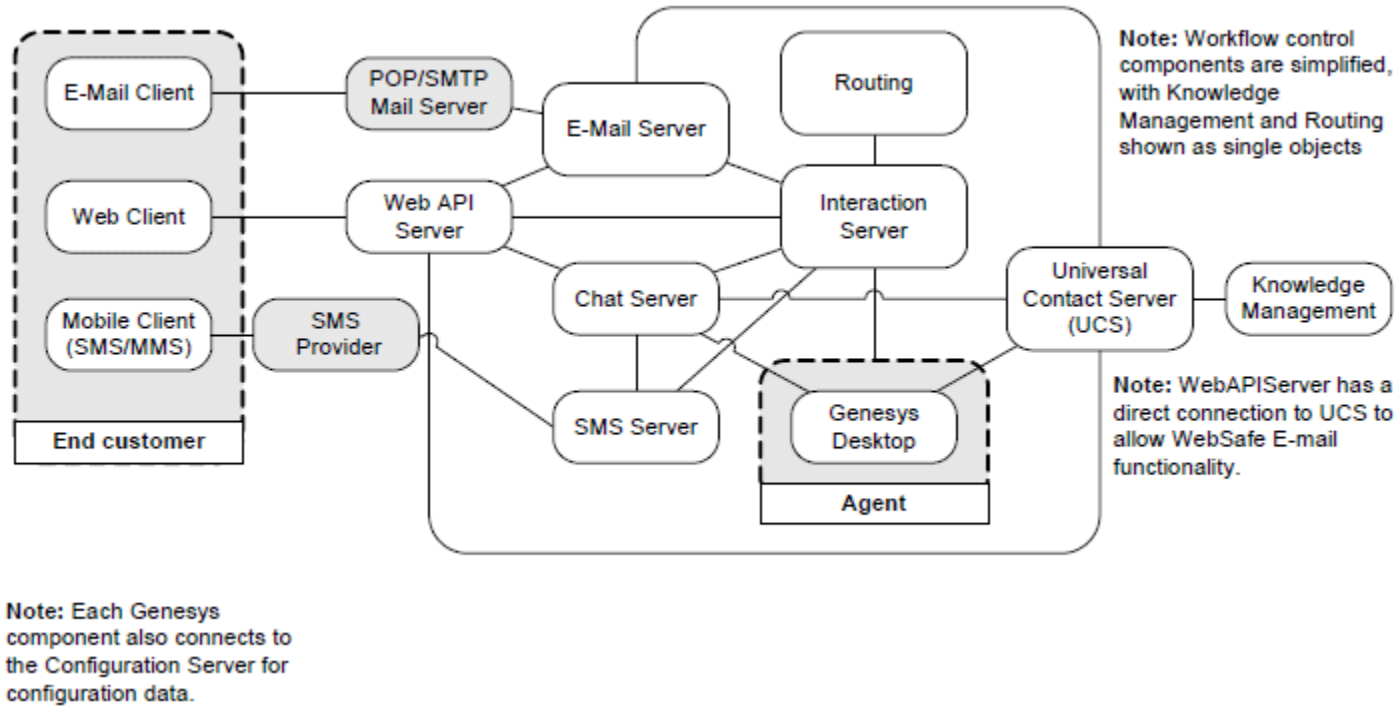
To summarize interaction flow:

- At the highest level the flow is controlled by interaction workflows that Interaction Server executes.
- Each interaction workflow contains queues and routing strategies.
- Routing strategies may bring in other applications/components to apply processing to the interaction—for example, sending a transcription of the chat session to the customer:
  - Send an acknowledgment or an automatic reply.
  - Apply a screening rule.
  - Apply content analysis (with Content Analyzer option only).
  - Forward or redirect the interaction.

For some media types (such as chat), the media interface also communicates directly with the agent desktop.

# Architecture

The figure below shows the main eServices components.




eServices Architecture

> **Important**
>
> For simplicity, not all connections and details are included in the figure. For example, many components (including Web API Server) connect to Stat Server.

Interaction processing generally proceeds according to the type of interaction. The following sections describe how five of the most common interaction types (email, chat, SMS, MMS, and web callback) are processed.

## Processing E-Mail

1. E-mail interactions arrive in one of two ways:
   - If the customer sends ordinary email, the interaction arrives via the enterprise mail server.

- If the customer sends email from a web site (by filling out a web form), the interaction arrives via the Web API Server.

2. E-mail Server stores the body of the interaction in the Universal Contact Server database, and then sends operational data on the interaction to Interaction Server.
3. Interaction Server parks the interaction's operational data in its cache and starts processing the data according to an interaction workflow.
4. What happens next depends on the interaction workflow and the routing strategies that it contains. The system may:
   - Apply a screening rule.
   - Assign the interaction to one or more categories (if Content Analyzer is present).
   - Generate an automatic response.
   - Route the interaction to an agent's desktop, possibly also sending an automatic acknowledgment to the customer.

   A supervisor may intervene at various points as long as the interaction's operational data remains in the Interaction Server's cache and the interaction is not being actively worked on by the Routing components.
5. The agent receives the interaction. The agent may then:
   - Simply reply to the interaction.
   - Reply making use of a standard response. With the Content Analyzer option, the interaction may have arrived already equipped with a category assignment and associated suggested response. Otherwise, the agent may search manually for a category with suggested response.
   - Transfer the interaction to another agent.
   - Produce a collaborative response by consulting with other agents.
   - Return the interaction to the system for further processing.
6. When the agent or agents finally release the reply (typically to an Outbound queue in the Interaction Server cache), the interaction workflow may route it to a senior agent or supervisor for QA review. The reviewer decides whether to let the reply continue through the outbound part of the interaction workflow, return it to the agent for revision, or take other action.

## Processing Chat

1. Chat interactions begin processing when the Web Client submits a customer's chat request to Chat Server.
2. Chat Server creates a chat session and asks Universal Contact Server to create an interaction record.
3. Chat Server submits the interaction to Interaction Server.
4. Interaction Server places the interaction in its initial queue and begins processing it according to an interaction workflow.
5. The interaction workflow and its component routing strategies may do various things, including sending a message to a customer prior to an agent actually handling the interaction, but eventually they select an agent who is available for chat sessions and send an invitation to that agent to participate in a chat session.

6. The agent connects to the chat session and accepts the invitation.
7. Agent and customer conduct a chat session (exchange with messages and notifications).
8. The chat session ends (by agent request).
9. Chat Server writes the content of the chat session to the Universal Contact Server database and updates the interaction in Interaction Server.
10. Any postprocessing occurs; for example, a transcript of the chat session is emailed to the customer.

## Processing SMS Messages

SMS messages arrive when a mobile client sends an SMS message to a phone number of a Contact Center. Genesys SMS Server is a recipient and handler of SMS messages.

SMS Server supports two operational modes:

**Paging mode** refers to receiving an individual SMS message from a mobile client and sending back an agent's response (paging inbound), or sending an individual SMS message to a mobile client on a Contact Center initiative (paging outbound).

**Session (chat) mode** refers to creating and keeping an interactive conversation between a mobile client and an agent in the form of a conventional chat session. All messages received and sent during this session are associated with one interaction, which corresponds to this SMS session.

### Paging Mode

Paging mode incorporates capabilities to send and receive individual SMS messages. SMS server:

- Submits the incoming SMS messages to Interaction Server as a new interaction.
- Sends SMS messages to mobile clients on requests from an agent, routing strategy, or application.

### Session Mode

This mode supports an interactive conversation between a client and an agent:

- For an incoming SMS message, SMS Server checks if the mobile client is participating in an active SMS session (Chat Server session).
- If a session is found, the SMS Server forwards the message to the session.
- If a session is not found, SMS Server creates a new SMS session. SMS Server:
  - Requests Chat Server initiate a chat session.
  - Stores a record about this session.
  - Starts forwarding incoming and outgoing messages between the mobile client and the Chat Server session.

## Processing MMS Messages

MMS messages arrive when a mobile client sends an MMS message to a phone number of a Contact Center. Genesys SMS Server is a recipient and handler of MMS messages. MMS messages can include a subject, text parts, multimedia parts, and a presentation scenario specified in an SMIL (Synchronized Multimedia Integration Language) part of an MMS message. SMS Server supports inbound mode for MMS messages.

Inbound mode refers to receiving an individual MMS message from a movable client and submitting it to Interaction Server as a new interaction.

## Processing Social Media

For information about processing Social Media, refer to the eServices Social Media Solution Guide.

# Deployment Planning

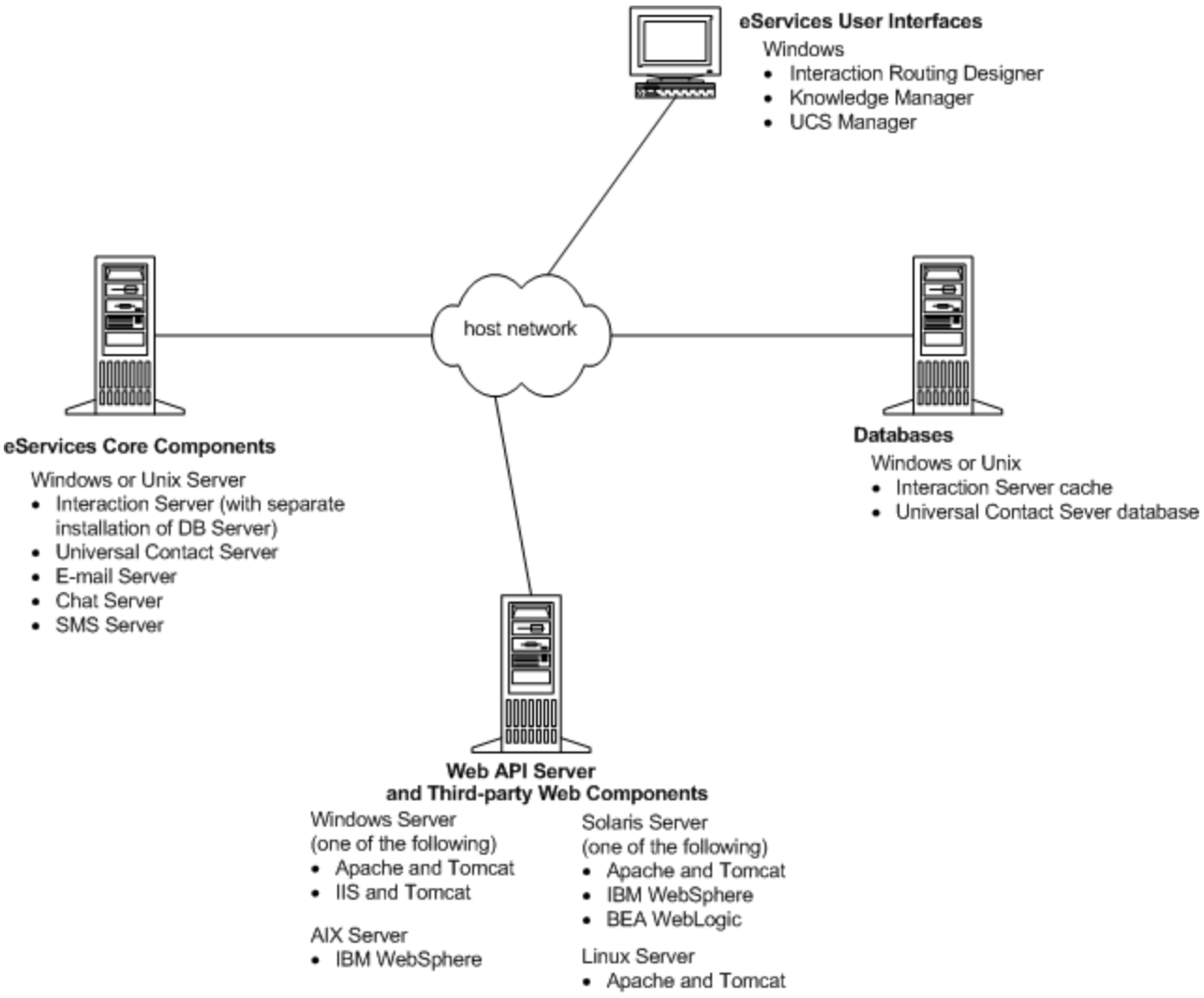
The Deploying on Windows and Deploying on UNIX sections, describe deployment in detail. Before beginning the deployment process there are several things you can think about and do to make the process easier.

## Overview

Genesys recommends that you configure eServices using the wizards provided with it and that, for Windows deployments, you install it using the integrated installation package. These utilities configure and install all eServices components. This section provides general information on deployment planning. For further information about specific issues, see also the eServices Administrator's Guide.

## Component Distribution

Genesys recommends that you divide eServices and associated components among several host machines.




See this note on Web API Server and Third-party Web Components.

When choosing host machines for your eServices deployment, keep the following recommendations in mind:

- Keep web servers for different applications on separate machines.
- Install eServices graphical user interface (GUI) applications on Windows hosts only. These GUI applications include:
  - Knowledge Manager
  - Universal Contact Server Manager
  - Interactions Workflow Samples
- Based on the load and nature of contact center media (e-mail, chat, or blended), you might need to deploy the following components on separate machines:
  - Universal Contact Server

- E-mail Server
- Chat Server
- SMS Server

Information about distributing eServices components across multiple hosts for load-balancing purposes is available.

- You should also deploy the corporate mail server on a separate computer.

> **Important**
> For low-level load configurations such as a functional lab environment, all components can be installed on a single host.

# Deployment Tasks

This section guides you through the various actions you must take to plan, deploy, and test your eServices solution for both Windows, and UNIX.

## Deployment Summary

### Windows

This section describes a model setup of a blended (e-mail and chat), single-tenant eServices solution deployed on a single Windows host machine, such as might be deployed in a lab setting. Complete these actions in the following order to deploy your solution.

## Plan your deployment

1. Choose which machines will host the eServices and various third-party components. For a sample architecture layout, see Component Distribution.
2. Choose which machines will host your databases. To help you calculate the rough size of your databases, use the formulas in Estimating Database Size.
3. Print out and complete the Configuration Worksheet with the values that fit your deployment.

## Complete all prerequisite actions

1. Verify that all Prerequisites components are in place.
2. Create the Host.
3. For deployments across multiple machines, Create a Shared Directory.
4. Create UCS and Interaction Server databases.

## Note on third-party software for processing Web-based interactions

Although Web API Server is not updated in the 8.5.0 release, you can use the 8.1.2 version with

eServices 8.5.0.

Web API Server requires a web server and an application container; deployment of some common combinations of those items is described in the "Deploying Third-Party Web Applications" chapter of the eServices 8.1 Deployment Guide.

## Configure eServices objects

1. Install the configuration wizards.
2. Create sample workflows for routing simple test interactions through the contact center to an agent.
3. Configure the eServices objects.
4. Configure Chat Server to communicate with Web API Server.
5. Create configuration objects and sample workflows for routing web callback interactions through the contact center to an agent. See Install Web Callback.

## Install eServices components

Launch the Integrated Installation for all eServices components

> **Important**
>
> For deployments across multiple hosts, you need to launch the Integrated Installation one time for each host.

## Deploy those eServices components not included with the wizard (Optional)

Manually deploy the eServices components that are not included in the eServices Configuration Wizard installation.

### Universal Contact Server Proxy or Interaction Server Proxy

1. Create the UCS Proxy and Interaction Server Proxy application objects.
2. Install UCS Proxy and Interaction Server Proxy.
3. Configure the UCS Proxy and Interaction Server Proxy Desktop applications.

### SMS Server

1. Create the SMS Server application object.
2. Install SMS Server.
3. Configure the eServices components.

## Verify Connections

1. Double-check that the connections you made with the configuration wizard are in place. For a list of necessary connections, see Connections.
2. Using Genesys Administrator Extension, add a connection from Stat Server to Interaction Server (or to Interaction Server Proxy). Stat Server uses this connection to find and route interactions to available eServices agents.

For more information, see Verifying connections.

## Modify the database

Run the database scripts to modify the database to process interactions.

## Test the Components

1. Test the eServices servers.
2. Test the components using the Web-based TestTool.
3. Test the Genesys Desktop.

## Test the Setup

1. Configure ABC Simple BP for routing e-mail interactions.
2. Configure ABC Simple Chat BP for routing chat interactions.
3. Send an e-mail to the Desktop.
4. Start a chat session.
5. Send a web form e-mail.
6. Configure WebCallback BP for routing web callback interactions.
7. Update the Interaction Server database for processing web callback interactions.

8. Send a web callback.

## UNIX

1. Prepare your environment
   a. Create a shared directory on your UNIX host that your Windows host can access.
      OR
      Create a shared directory on the Windows host that your UNIX hosts can access.
   b. Create your databases.
2. Launch the eServices Configuration Wizard from a Windows host to configure the components that will run in the Solaris, Linux, or AIX environments.
   a. Install the Configuration Wizards.
   b. Install the Interaction workflow samples.
   c. Configure the objects.
3. Run the installation package for each remaining component:
   `<component>\<platform>`
4. (Optional) Manually deploy the eServices components that are not included in the eServices Configuration Wizard installation.
   - Universal Contact Server Proxy or Interaction Server Proxy
     a. Install UCS Proxy.
     b. Install Interaction Server Proxy.
   - SMS Server: Install SMS Server.

# Using the eServices Configuration Wizards

> **Important**
>
> This document is currently being updated as eServices Configuration Wizards are no longer supported. You must install the eServices components using their Installation Packages separately. For information on how to do manual installation of the components, see Manual Installation of eServices Components.

The eServices Configuration Wizards consist of a main wizard and multiple secondary wizards, which are launched automatically when you indicate that you want to configure particular components. The wizard presents only those pages that are applicable for your deployment, or for the stage of deployment that you have reached.

This section summarizes the flow (including installing the wizards and workflow samples) of the main and secondary wizards to deploy the model setup. The information in the Key Actions and Inputs (Model Setup)column gives you the model-specific data inputs and actions for that section of the wizard.

The details of your solution configuration are entered into the Configuration Database when you have completed the wizard. Exiting the wizard prior to completion requires that you begin the solution configuration from the beginning. Any components you create along the way, however, are available during subsequent solution configuration using the wizard.

## Configuration Wizards

## Using the Wizard to Configure your eServices Components

## Installing the Configuration Wizards

### Prerequisites

- Interaction Management CD.
- Start the Configuration Layer.

1. Install the eServices Configuration Wizards and Genesys Wizard Manager.
   a. In the root directory of the Interaction Management CD, find `Setup.exe`.

   b. Run `Setup.exe` and follow the directions. You may want to read the Wizard Advisory supplied with the wizard.

2. If you intend to configure your Stat Server as you configure eServices, you should also install the configuration wizard for this product.

## Installing and Starting the Workflow Samples

Interaction Workflow Samples use a number of preconfigured Script objects of various subtypes, including Simple Routing, Interaction Queue, Workbin, and Interaction View.

### Prerequisites

- Interaction Management CD.

1. Install the Interaction Workflow Samples.
   a. On your Interaction Management CD, find and double-click `Setup.exe` in the `solution_specific\InteractionWorkflowSamples\windows` directory.
   b. At the welcome page for the installation wizard, click **Next**.
   c. Specify the destination for Interaction Workflow Samples, and click **Next**.
   d. Click **Install**.
   e. Click **Finish**.
2. Start the Interaction Workflow Samples.
   a. From the Windows taskbar, select **Start > Programs > Genesys Solutions > eServices 8.1.3 > Interaction Workflow Samples > Start Interaction Workflow Samples**. This launches the Interaction Workflow Samples Wizard.
   b. Login to the wizard using your user name and password.
   c. At the Welcome to Interaction Workflow Samples Wizard window, click **Next**.
   d. If using a multi-tenant environment, select the Tenant and click **Next**.

> **Tip**
> During the installation, the setup procedure does not look for possible name conflicts between existing objects and new components from the Interaction Workflow Samples. It overwrites any existing objects. In order to prevent the loss of existing objects, Genesys recommends you install the new samples into a separate tenant. Alternatively, you should use IRD's Business processes export capability to create backups of workflows and strategies related to an existing configuration.

e. Specify a destination directory for the strategy files, and click **Next**.
You may want to select the directory used for your Interaction Routing Designer strategies, unless it already contains strategies with same file names, which would then be overwritten. If you do decide to use that particular directory, make a backup of its content before proceeding.

> **Tip**
> Do not select the `StrategyFiles` directory created during installation of your Interaction Workflow Samples as the target here. This causes a file "collision", since the source and target files are then identical.

f. At the **Import Completed** window, click **Next**.

Click **Finish**. Once you exit the wizard, you can view the new objects (over 180 of them) installed with these Samples, by opening the `<tenant>\Scripts` folder in Configuration Manager or Genesys Administrator.

## Launching the Configuration Wizard

To launch the Genesys Wizard Manager:

1. From the Windows taskbar, select **Start > Programs > Genesys Solutions > eServices 8.1.0 > eServices Configuration Wizards > Start Wizard Manager**.
2. Click **Log into the Configuration Layer**. This opens the main **Genesys Wizard Manager** window.
3. Select **Multimedia** from the menu on the left side of the window, and then select **Deploy Multimedia Solution in your contact center**. The **eServices Configuration Wizard Welcome** page opens.
4. Click **Next** to begin configuring eServices objects.

## Naming your Solution

- At the **Solution Name** page, enter a name for your solution and click **Next**.

```
Name = ES85
```

## Selecting the Configuration Type

Select Simple or Custom Configuration

1. At the **Configuration Process Selection** page, select one of the following:
   - Simple single-host configuration
   - Custom Configuration

> **Important**
> For this model setup, select **Custom Configuration**.

2. Click **Next**.

> **Important**
> The **Simple single-host configuration** option is available for a predefined host in Windows deployments only. This chapter describes the custom configuration process. If you choose the simple configuration with Windows, use this chapter by skipping those steps that the wizard does not present to you during deployment.

## Copying the IP to your shared directory

1. At the **Installation Package** page, click **Have Disk** and navigate to the Interaction Management CD.
2. To select a destination for the package, click **Browse** and navigate to your shared directory).
3. Click **Next**.

> **Important**
> eServices uses this group of components for installation. In later steps you will copy to this directory installation packages that are specific to each eServices component.

4. At the **Installation Ready** page, confirm that your software is ready for installation and click **Next**.

## Selecting or Adding Message Server

1. At the **Solution Components: Message Server** page, select or add a Message Server.

2. Click **Next**.

## Creating the Component DAP

At this point in the wizard, you begin creating your DAPs and associated DB Servers. You must create all DAPs and DB Servers here, before moving on to other components. The eServices Configuration Wizard does not permit you to create DAPs later in the configuration. Relaunch the Database Access Point Wizard as necessary.

1. At the **Solution Components: Data Access Point** page, click **Add**.
2. At the **Browse for Application** page, click the **New Application** icon.
3. Create a new DAP.
4. Enter your Database Access Point information:

| Component DAP | Application Name | DB Server Name | JDBC Connection | Database Information | Case Conversion |
|---|---|---|---|---|---|
| Interaction Server | ES85_Ixn_DAP | ES85_IxnDBServer | Clear the check box<br>**Note:** It is important to clear the Enable JDBC access box. Failing to do so can lead to a configuration problem. | • DBMS Type = mssql<br>• DBMS Name = ESHost<br>• Database Name/SID = IxnDB<br>• User Name = sa<br>• Password = <password for user sa> | any<br>**Note:** The Interaction Server DAP must have its Case Conversion attribute set to any or upper. Setting it to lower causes an error when Interaction Server initializes. |
| Universal Contact Server | ES85_UCS_DAP | [NONE]<br>**Note:** UCS connects to its database | Select the check box, and enter the following information:<br>• Host = ESHost | • DBMS Type = mssql<br>• DBMS Name =""<br>**Note:** If | any |

| Component DAP | Application Name | DB Server Name | JDBC Connection | Database Information | Case Conversion |
|---|---|---|---|---|---|
| | | directly through JDBC. You do not need to create a DB Server Application object for it. | • Port (for JDBC) = 1433 (Microsoft SQL default)<br>• Role = Main<br>**Note:** If your RDBMS is Microsoft SQL Server on a different host, select the host where you have Microsoft SQL Server installed, and enter the connection information for that instance of the database. | you do not clear the DBMS Name box, your UCS will not work properly.<br>• Database Name/SID = Customer<br>• User Name = sa<br>• Password = <password for user sa> | |

5. Click **Finish** to complete the Database Access Point Wizard and return to the eServices Configuration Wizard.
6. In the **Browse for Application:** page, select the Database Access Point that was just configured and click **OK**.

> Tip

If you are using a Microsoft SQL 2005 database, an additional configuration step is recommended for the UCS DAP.

In Configuration Manager or Genesys Administrator, create a `settings` section on the **Options** tab. Create a new option, setting the option name as `prepare` and the option value to **false**.

Additional details are provided in the eServices Administrator's Guide.

## Adding DB Server

1. Click the folder icon beside DB Server, and then click the **New Application** icon to run the DB Server Wizard.
2. Follow the wizard's directions and enter the following information:

| Application Name | Host | Default Port | DBMS Type |
|---|---|---|---|
| ES85_IxnDBServer | ESHost | 6110 | mssql |

3. At the **Installation Package** page, copy the DB Server installation package. Select a source (the Management Framework CD) and a destination (the shared directory). Click **Next**.
4. At the **Installation Ready** page, confirm that your DB Server software is ready for installation, and then click **Next**.
5. At the **Listening Ports and Transport Layer Security (TLS) Settings** page, click **Next**.

> **Important**
>
> For this model setup, we do not require any additional ports. If you want to install additional ports, you can do so in Configuration Manager or Genesys Administrator later.

6. At the **Log Configuration** page, take the defaults. Click **Next**.
7. At the **Backup Server Information** page, because no backup servers are configured, clear the checkbox and click **Next**.
8. Click **Finish** to exit the DB Server Wizard. The Database Access Point Wizard will resume.

## Adding UCS and Interaction Server

1. At the **Solution Components:** page, select your component, and Click **Add**.
2. At the **Browse for Application :** page, click the **New Application** icon.
3. Follow the wizard's direction, and enter the following information:

| Application Name | Type | Host | Default Port | API Port | Connections | License Connection | Login Account |
|---|---|---|---|---|---|---|---|
| ES85_UCS | N/A | ESHost | 6120 | Accept the default port value, or enter a port number where UCS should listen for third-party protocol connections. | • Message Server<br>• DAP: ES85_UCS_DAP | N/A | Select the user account (Configuration Layer Person object) or Access Group that UCS is to use to log in to the Configuration Layer. UCS uses the Configuration Layer to pass some of its information back and forth to certain components. The selected account or access group must have write |

| Application Name | Type | Host | Default Port | API Port | Connections | License Connection | Login Account |
|---|---|---|---|---|---|---|---|
| | | | | | | | access to the tenant in use. Refer to Configuration Server access permissions for more information. |
| ES85_IxnSrv | Select New-style.<br>• The eServices Configuration Wizard automatically creates a multimedia Switch object for you in the background. Continue at "Configure Framework Resources".<br>• If your configuration does not include a properly configured Multi | ESHost | 6130 | N/A | • Servers for Third-Party Protocol<br>• DAP: ES85_Ixn_DAP.<br>If you already installed related eServices components (Universal Contact Server, E-mail Server, and Classification Server, for instance), you can connect | • License Server Host = ESHost<br>• License Server Port = 7260<br>• Specify the number of licenses for Interaction Server features | N/A |

| Application Name | Type | Host | Default Port | API Port | Connections | License Connection | Login Account |
|---|---|---|---|---|---|---|---|
| | media type Switch object, continue at "Create a Multimedia Switch object (Switch-based Interaction Server only)" | | | | to them now using this page. | | |

4. At the **Log Configuration:** page, accept the defaults and click **Next**.
5. At the **Installation Package:** page,
   - Click **Have Disk**, navigate to the Interaction Management CD, and then click **OK**.
   - Click **Browse**, navigate to your shared directory, and then click **Next**.
6. At the **Installation Ready:** page, click **Next**.
7. Click **Finish** to exit the Wizard and return to the eServices Configuration Wizard.
8. At the **Browse for Application:** page, select the the server and click **OK**.
9. Click **Next**.

## Creating a Multimedia Switch Object (Switch-based Interaction Server only)

> **Important**
>
> Before attempting to run your eServices solution, check Configuration Manager or Genesys Administrator for the existence of a `Multimedia Switch` object. If for some reason the wizard has failed to create one, use the Framework Wizard to create it. No other configuration is required; the components that require this switch are able to locate it automatically.

1. If you select **Switch-based Interaction Server**, the wizard checks for the following:
   - A `Multimedia`-type switch
   - A connection from the switch to a `T-Server` type Interaction Server

   If the wizard fails to detect either the switch or its connection to T-Server, it issues a warning. Click **Proceed** to launch the Switch Wizard.

   > **Important**
   > You can use a preexisting Switch object (that is connected to a properly configured Interaction Server) from your Configuration Layer, if available.

2. To create a new `Multimedia` type Switch object, provide a unique name for it and click **Next**. (The wizard allows you to create the new Switch only in folders designated to contain objects of type Switch.)
   **Name** = `ES85_Switch`

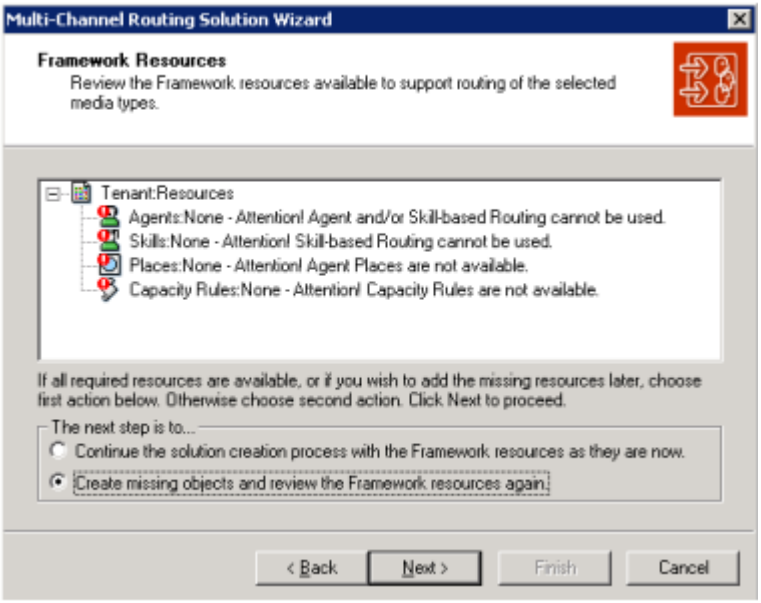
   > **Important**
   > Even though the Configuration Layer permits assignment of multiple Switches of type Multimedia Switch to a single Interaction Server Application object, Genesys does not currently support such a configuration. This type of configuration can lead to unpredictable behavior on the part of the application that is connected to the switches.

3. At the **Switching Office:** page, click **Add** to create a new Switching Office object. Enter the following information:
   - **Name** = `ES85_Switching_Office`
   - **Switch Type** = `Multimedia Switch`
4. Click **Finish**.
5. At the **Switch Summary** page, review the details and click **Next**.

## Configuring Framework Resources

1. If you select `New-Style Interaction Server`, the wizard opens the **Framework Resources**. This page lists the following:
   - Current Framework resources

- Any missing resources required to support the media types served by your switch




2. To create the missing resources now, select **Create missing objects and review the Framework resources again**, and then click **Next**. For this model setup, select this option.
   In your own deployment, you can choose to wait until after you finish the wizard to create the required objects. Select **Continue the solution creation process with the Framework resources as they are now**.
3. At the **Framework Objects** page, select the object type you want to create from the drop-down menu. For this model setup, create and configure the various objects as follows:

| Object Type | Model Setup |
|---|---|
| Place | Create `Place` objects to facilitate the routing of multimedia interactions:<br>• **Place Name** = `Agent1Place`<br>**Note:** Do not create any DNs associated with this `Place` object, though you may want to create a `Place Group` object for this place, as suggested in the wizard. |
| Capacity Rule | By default, as the final step of the solution's configuration, the<br>wizard creates and installs three capacity rules for your tenant. Do not create any capacity rules here |

| Object Type | Model Setup |
| --- | --- |
| | (although you can, if you have previously installed the Resource Capacity Wizard, which is available on the Real-Time Metrics Engine CD). |
| Agent | Create `Person` objects to represent agents in your model setup<br><br>and assign agent IDs to them. (Also use this step to create any `Person` objects you intend to use as the autoreply agents for supplying agent-related information in standard responses).<br><br>• **Agent Name** = `Agent One`<br>• **Employee ID** = `Agent1`<br>• **User Name** = `Agent1`<br>• **Password** = `Agent1`<br><br>**Note:** The agent values supplied here are for general use in this model setup, and do not correspond to the actual steps suggested by the wizard for the creation of `Person` objects. A `Person` object's user name must be unique within the Configuration Database. This is not a concern for this model installation, which is in a single-tenant environment. If you were installing more than one set of eServices components in a multi-tenant environment, you would use distinct names across tenants. |
| Agent Login | Create `Agent Login` objects to enable agents to log in to the |

| Object Type | Model Setup |
| --- | --- |
| | switch. The agent login code must correspond to the agent login numbering of your switch. |
| Skill | You do not need to create skills for this model installation. |

## Selecting a Stat Server

At the **Solution Components:** page, select **Stat Server:**. Click **Add** and select a Stat Server. This connects all the appropriate components in your eServices solution to this Stat Server.

> **Important**
>
> Even though the wizard allows you to create a Stat Server here, do not do so. Prior to using the wizard, you should have one Stat Server per eServices tenant already configured and installed in your environment.

If you receive a warning message, it indicates that Stat Server does not yet have a connection to Interaction Server (or to Interaction Server Proxy). This connection is required for both the routing and reporting of eServices interactions.

- If you click **Yes**, you may receive an error message if the Stat Server wizard is not installed on your machine. The connection cannot be created at this time.
- If you click **No** (the recommended approach), make sure that you add the connection from Stat Server to Interaction Server before you attempt to use routing or reporting with the eServices solution.

## Adding the Other Components

1. At the **Solution Components:** page, select your component, and Click **Add**.
2. At the **Browse for Application :** page, click the **New Application** icon.
3. Follow the wizard's directions, and enter the following information:

| Component | Application Name | Host | Default Port | Connections | Other Details |
| --- | --- | --- | --- | --- | --- |
| Universal Routing Server | Select the URS that you | | | | |

| Component | Application Name | Host | Default Port | Connections | Other Details |
|---|---|---|---|---|---|
| (URS) | created and installed as one of the prerequisites for your eServices installation. If you receive a warning but stating that URS is not configured to monitor Interaction Server, click **Yes** to reconfigure now, or click **No** to reconfigure later. If you click **No**, make sure you add the required connections after configuration and installation are complete. | | | | |
| Application Cluster | Add an application cluster if you intend to group your eServices components into groups for load-balancing purposes. After giving a name to your new application cluster, the Application Cluster Wizard invites you to group, for instance, Chat Server, and E-mail Server, as well as | | | | |

| Component | Application Name | Host | Default Port | Connections | Other Details |
|---|---|---|---|---|---|
| | other application clusters. | | | | |
| Web API Server | | | | | Although it may appear in the Wizard, this component is not included in the 8.5.0 release. |
| Universal Contact Server Manager | ES85_UCSMgr | | | Universal Contact Server | |
| E-mail Server | ES85_E-mailServer | ESHost | 6150 | • Message Server<br>• Interaction Server<br>• UCS | • POP3 server name = ESHost<br>• POP server type = POP3<br>• SMTP server name = ESHost<br>**Note:** This model installation uses ESHost as its POP3 and SMTP servers. If you are using your enterprise POP3 server, enter its name as the POP3 server |

| Component | Application Name | Host | Default Port | Connections | Other Details |
|---|---|---|---|---|---|
| | | | | | name value. You do not need to enter a value for the SMTP server name if it is the same server that you used for POP3 server name.<br>• Address = <mailboxName>@ESHost<br>• User name = <username><br>• Password = <password><br>• E-mail address default domain name = ESHost<br>• External Agent e-mail address = external@ESHost<br>• Enable Web Form Processing by selecting the check box, and provide a port where |

| Component | Application Name | Host | Default Port | Connections | Other Details |
|---|---|---|---|---|---|
| | | | | | the Web API Server should listen for requests—for example IWF Processing Port = 7777.<br>• Enter the e-mail address that is used to supply field values in automated responses generated in the contact center. Select an existing Person object from the Configuration Layer:<br>Auto-reply Agent = Agent1<br>• Configure e-mail accounts that exist on your corporate e-mail server for customers to use when contacting your company—for example: |

| Component | Application Name | Host | Default Port | Connections | Other Details |
|---|---|---|---|---|---|
| | | | | | • Customer access account 1 = <mailboxName1>@ES<br>• Customer access account 2 = <mailboxName2>@ES<br>**Note:** If you installed the Interaction Workflow Samples, then your configuration will automatically include the following three e-mail accounts (which are used in the samples): Customer support, Tech support, and Warranty support. If you did not install the Interaction |

| Component | Application Name | Host | Default Port | Connections | Other Details |
|---|---|---|---|---|---|
| | | | | | Workflow Samples before running the wizard, then no e-mail accounts are shown. You must add and configure at least one e-mail account before continuing. |
| Classification Server | ES85_ClassificationServer | ESHost | 6160 | • Message Server<br>• UCS | • License Server Host = ESHost<br>• License Server Port = 7260 |
| Training Server | ES85_TrainingServer | ESHost | 6170 | • Message Server<br>• UCS | |
| Knowledge Manager | ES85_KnowledgeManager | | | • Message Server<br>• UCS | Indicate that you have a Classification Server Content Analysis license installed by selecting the check box of the same name. If you do not have this license, |

| Component | Application Name | Host | Default Port | Connections | Other Details |
|---|---|---|---|---|---|
| | | | | | do not select this check box. |
| Chat Server | ES85_ChatServer | ESHost | 6180 | • Message Server<br>• Interaction Server<br>• UCS | You can change the ESP and webapi ports if necessary. ESP refers to the ability of URS to submit messages directly from a strategy into a chat session—for example, to greet a customer who is waiting for an agent. The webapi port is to be used by Web API Server. |

4. At the **Log Configuration:** page, accept the defaults and click **Next**.
5. At the **Installation Package:** page,
   - Click **Have Disk**, navigate to the Interaction Management CD, and then click **OK**.
   - Click **Browse**, navigate to your shared directory, and then click **Next**.
6. At the **Installation Ready:** page, click **Next**.
7. Click **Finish** to exit the Wizard and return to the eServices Configuration Wizard.
8. At the **Browse for Application:** page, select the the server and click **OK**.
9. Click **Next**.

## Creating Capacity Rules

1. At the **Resource Capacity Rules Deployment:** page, select a folder in the Configuration Server where eServices should store its resource capacity rules. Click **Next**.
2. At the **Resource Capacity Rules Selection:** page, select the capacity rule you want to use as the default for eServices. The wizard assigns that rule to the Tenant object associated with your deployment in the previous step. Click **Next**.

## Completing the Wizard

1. At the **Solution Summary:** page, review solution components you have collected for use with eServices. Click **Next** to finish the configuration or **Back** to change the configuration of some solution component.
2. At the **Completing the eServices Solution Wizard:** page, click **Finish** to confirm your solution configuration. Remember that you can change configuration details at a later time as well.

> **Important**
>
> Only at this point, when you exit the wizard, are the details of your solution configuration entered into the Configuration Database. Exiting the wizard prior to this point requires that you begin the solution configuration from the beginning. Any components you create along the way, however, are available during subsequent solution configuration using the wizard.

# Estimating Database Size

This section suggests some ways of estimating the size of the Universal Contact Server (UCS) and Interaction Server databases. Please keep in mind that these are rough estimates only. The size of a database file on a hard drive depends on a number of factors other than the number of interactions it contains. For example:

- The size of a database file can vary according to the DBMS platform you are using.
- Interactions are not uniform in size. An e-mail may be as small as 4 KB or it may be 200 KB or more. Also, interactions can include attachments, which can increase size greatly.

> **Important**
>
> You can configure Universal Contact Server Manager to automatically archive and/or prune old messages to keep this database to a manageable size. See Universal Contact Server Manager Help for more information.

Interaction Server's Event Logger functionality requires its own separate database.

The following procedures describe how to estimate the amount of disk space require for the eServices databases.

## Estimating UCS database size for e-mail interactions

## Estimating UCS database size for e-mail interactions

1. Multiply the number of interactions expected each day by the number of days you want to keep the interactions.
2. Multiply the result by 25 KB (estimating 20 KB for the message body and 5 KB for headers, comments, and indexes).
   For example, if you anticipate 100 incoming interactions per day and you want to keep 90 days' worth of history, reserve 100 x 90 x 25 KB or 225 MB of disk space.

> **Important**
>
> Though the average interaction will probably be less than 25 KB in size, this is a reliable estimate of the amount of disk space required for contact and history records.

3. Add 1 KB for each related contact in the database. So for 100,000 contacts, allow 100 MB for the database.

## Estimating UCS database size for chat interactions

## Estimating UCS database size for chat interactions

1. Multiply the number of chat interactions expected each day by the number of days you want to keep the interactions.
2. Multiply the result by 5 KB. For example, if you anticipate 1000 incoming chat interactions per day, and you want to keep 90 days' worth of history, reserve 1000 x 90 x 5 KB, or 450 MB of disk space.
3. Add 1 KB for each related contact in the database. So for 100,000 contacts, allow 100 MB for the database.

## Estimating the size of the Interaction Server database

## Estimating the size of the Interaction Server database

1. Use the following formula:
   $(3.5 + d)m = x$ KB
   Where:
   - 3.5 KB is the size of system data and business properties (also called operational data). See the initialization script for the particular limit for each property.
   - d is the size, in KB, of attached user data. Attached data is normally 4 KB or less, but it may be more. eServices imposes an upper limit on the size of attached user data written in the logs. However, there is no limit set for the total amount of attached data that Interaction Server has to process. Further information about the Interaction Server database is available in the eServices Administrator's Guide.
   - m is the expected maximum number of active interactions. In choosing this number for Genesys E-mail, take into consideration the large number of e-mails that may build up during off hours or system malfunctions. For example, if you limit attached user data to 64 KB and you expect a maximum of 10,000 active interactions, allow about 700 MB for the database.

# Configuration Worksheet

Before you begin deployment, it is helpful to assemble information that you will need to supply.

Use the worksheet to plan your eServices deployment.

This worksheet lists:

- Values (mostly Application object names and port numbers) that you must invent. You can invent them ahead of time in case you want to incorporate some systematically (for example, you may want all Application object names to start with a certain prefix, or you may want port numbers to be consecutive or to all start with the same two digits).
- Example values.
- Names (or other attributes) of non-eServices objects to enter. Some of these are likely to exist already. Of those which do not, you can create them before or after installing. For example,
  - eServices requires several databases (tablespaces): one for Interaction Server and two for UCS.
  - eServices requires a Person or Access Group object that UCS can use to log in to Configuration Server.

## Completing the Worksheet

## Completing the Configuration Worksheet

1. Print out the worksheet.
2. Add deployment information for your environment in the Your Value column.
3. Keep a copy of this worksheet with you during the deployment process.

### Next Steps

- Continue with Deploying eServices on Windows.

## Configuration Worksheet

# Configuration Worksheet

| Field | Your Value | Model Setup Value | Description |
|---|---|---|---|
| **General** | | | |
| User name | | | User name for logging into Configuration Server. |
| User password | | | Password for logging into Configuration Server. |
| Application | | | Application object name—for example, `default`. |
| Host | | | Host of Configuration Server |
| Port | | | Port of Configuration Server host where Configuration Server listens for queries—for example, 2020. |
| **Solution** | | | |
| Name | | ES85 | Invent a name for your eServices solution. |
| **Tenant (in multi-tenant environment)** | | | |
| Name | | | Name of the Tenant object holding the eServices solution that you are configuring. |
| **Databases and Associated Configuration Objects**<br><br>**Note:** Before installing eServices, you must configure and install a DBMS and create four databases/tablespaces, one for Interaction Server and two for Universal Contact Server (one main, one archive). | | | |
| **Interaction Server Database Objects** | | | |
| **Database Access Point (DAP)** | | | |
| DAP Name | | ES85_Inx_DAP | Invent a name for the DAP to use with Interaction Server. If you intend to use the Event Logger or message queue functionality you |

| Field | Your Value | Model Setup Value | Description |
|---|---|---|---|
| | | | will need additional DAPs. |
| **DB Server** | | | |
| Name | | ES85_IxnDBServer | Name of the DB Server application object. |
| DBMS type | | Microsoft SQL Server | Type of database management system you are using:<br>• Microsoft SQL<br>• Oracle<br>• DB2 |
| Host | | ESHost | Host where you want DB Server installed. |
| Port | | 6110 | Port where DB Server listens for queries. |
| SNMP Port | | N/A | Port where DB Server listens for management commands (optional). |
| DBMS name | | ESHost | Name of your database management system. |
| Database name | | InxDB | Name of the database tablespace for Interaction Server. |
| User name | | sa | User name for logging into the DBMS. |
| Password | | <password for user sa> | Password for logging into the DBMS. |
| **Universal Contact Server (UCS) Database Objects** | | | |
| **DB Server** | | | |
| Not needed because UCS used JDBC to connect to the database. | | | |
| **Database Access Point (DAP)** | | | |
| DAP name | | ES85_UCS_DAP | Invent a name for the DAP to use with |

| Field | Your Value | Model Setup Value | Description |
|---|---|---|---|
| | | | UCS. |
| Host | | ESHost | Host name where the database s running. |
| Port | | 1433 | Port where the database listens for queries (for example, 1521 is the default for Oracle, 1433 for Microsoft SQL, 50000 for DB2). |
| DBMS type | | Microsoft SQL Server | Type of database management system you are using (Microsoft SQL, Oracle, DB2). |
| Database name | | Customer | Name of the database tablespace for UCS. |
| User name | | sa | Username for logging into the UCS DBMS. |
| Password | | <password for user sa> | Password for logging into the UCS DBMS. |
| **Universal Contact Server** | | | |
| Name | | ES85_UCS | Invent a name for the UCS application object. |
| Host | | ESHost | Host where you want UCS installed. |
| Port | | 6120 | Listening port for requests. |
| Universal Contact Server API port | | Default port value | Port for Remote Method Invocation (RMI) connection to the UCS API. Must be different from stander server port. |
| Login account | | N/A | The User Account (Person) or Access Group that UCS will use to log in to Configuration Server.<br>**Note:** This Person or Access Group must have |

| Field | Your Value | Model Setup Value | Description |
|---|---|---|---|
| | | | Administrator or higher access rights so it can update information in the Configuration Database. Use an existing account with these rights or create one for the purpose. |
| **Switching Office** | | | |
| Name | | | Select or create a multimedia switch.<br><br>**Note:** Include only one multimedia switch per tenant to preserve Stat Server and URS support for Interaction Server. |
| **Multimedia Switch** | | | |
| Name | | | Select or create a switching office. |
| **Interaction Server** | | | |
| Name | | ES85_InxSrv | Invent a name for your Interaction Server application object. |
| Host | | ESHost | Host where you want Interaction Server installed. |
| Port | | 6130 | Port where Interaction Server listens for queries. |
| **Stat Server** | | | |
| Name | | | Select or create a Stat Server. |
| Host | | | Host where you want Stat Server installed. |
| Port | | | Port where Stat Server listens for queries. |
| **Application Cluster** | | | |
| Name | | N/A | Invent a name for your Application Cluster. |
| **Web API Server** | | | |
| Not included in eServices 8.5.0. However you can use Web API Server 8.1.2 with eServices 8.5.0. | | | |
| **UCS Manager** | | | |

| Field | Your Value | Model Setup Value | Description |
|---|---|---|---|
| Name | | ES85_UCSMgr | Invent a name for your UCS Manager application object. |
| **E-Mail Server (for Genesys E-Mail)** | | | |
| Name | | ES85_E-mailServer | Invent a name for your E-mail Server application object. |
| Host | | ESHost | Host where you want E-mail Server installed. |
| Port | | 6150 | Listening port for requests. |
| POP server | | ESHost | The name of your corporate POP server. |
| SMTP server | | ESHost | The name of your corporate SMTP server if different from the POP server. |
| POP3 user name and password | | N/A | Address, user name and password for logging on to your POP server. |
| Default e-mail address domain | | ESHost | Domain to be appended to e-mail (both incoming and outgoing) that does not already have a domain specified in its To address field. |
| External agent e-mail address | | external@ESHost | Address to be used as the From address in messages to external agents and as the To address in messages from external agents. Typically, this is different from your general corporate e-mail addresses. |
| Integrated Web For processing port (optional) | | 7777 | Listening port number for Web Form-processing entry point. |
| Automated Reply Agent | | Agent1 | Notional agent name used in automatic |

| Field | Your Value | Model Setup Value | Description |
|---|---|---|---|
| | | | responses. Typically this is something generic like `Genesys Customer Care`. |
| E-mail account(s) for customer access | | <mailboxName1>@ESHost<br><mailboxName2>@ESHost | Account name and e-mail address. These are the accounts on your corporate POP server that E-mail Server will pull e-mails from and bring them into the eServices system. Be sure to also create them on your corporate mail server if the do not exist. |
| **Classification Server** | | | |
| Name | | ES85_ClassificationServer | Invent a name for your Classification Server application object. |
| Host | | ESHost | Host where you want Classification Server installed. |
| Port | | 6160 | Listening port for requests. |
| **Training Server** | | | |
| Name | | ES85_TrainingServer | Invent a name for your Training Server application object. |
| Host | | ESHost | Host where you want Training Server installed. |
| Port | | 6170 | Listening port for requests. |
| **Knowledge Manager** | | | |
| Name | | ES85_KnowledgeManager | Invent a name for your Knowledge Manager application object. |
| **Chat Server (for Genesys Chat)** | | | |
| Name | | ES85_ChatServer | Invent a name for your Chat Server application object. |

| Field | Your Value | Model Setup Value | Description |
|---|---|---|---|
| Host | | ESHost | Host where you want Chat Server installed. |
| Port (ID=default) | | 6180 | Listening port for requests. |
| Port (ID=webapi:recommended) | | 6181 | A specific port for web client connections only. |
| Port (ID=ESP) | | 6182 | Listening port for ESP requests. |
| ESP default nickname | | system | Name to be used in place of agent's name for use in automated chatting. |

# Connections Table

The following table lists connections that you set in the Configuration Layer for eServices components. eServices 8.1 supports Transport Layer Security (TLS) and client-side port functionality for some connections.

In addition to the information in the table, keep in mind that:

- Connections to Message Server are required only if you are planning to use it for application log output.
- A Local Control Agent (LCA) runs on each host. All Genesys components on a host connect to the local LCA. Refer to the Framework Deployment Guide for details on configuring LCAs.
- For Reporting purposes, you must also add a connection from your Reporting Stat Server to Interaction Server (or to Interaction Server Proxy).

> **Important**
>
> It is possible to have multiple connections to some components. In the table, these components are marked with a *.

| Application | Connects to |
|---|---|
| Chat Server | Interaction Server<br>Message Server<br>Universal Contact Server |
| Classification Server | Message Server<br>Universal Contact Server |
| E-mail Server | Interaction Server<br>Message Server<br>Universal Contact Server |
| Interaction Server | Chat Server (via the ESP port)*<br>Classification Server*<br>E-mail Server*<br>Interaction Server DAP*<br>Message Server<br>Digital Messaging Server/Social Messaging Server<br>Stat Server (for reporting)* (see Tip below)<br>Universal Contact Server*<br>Application Cluster* |
| Knowledge Manager | Universal Contact Server |
| Digital Messaging Server/Social Messaging Server | Message Server<br>Chat Server*<br>Interaction Server<br>Solution Control Server |

| Application | Connects to |
| --- | --- |
| | Universal Contact Server (Optional, depending on configuration) |
| Training Server | Message Server<br>Universal Contact Server |
| Universal Contact Server | Message Server<br>Stat Server*<br>UCS DAPs |
| Universal Contact Manager | Universal Contact Server |
| Web API Server | Not included in release 8.5.0 |

> **Tip**
>
> When Stat Server is configured in Hot Standby mode, Interaction Server must have connections to both Primary and Backup Stat Servers

# Web API Server

Starting in the 8.5 release, Web API Server has its own separate documentation.

# Third-Party Applications

Java Software Development Kit (JDK) or Java Runtime Environment (JRE) is required for various eServices functionalities. To deploy these, determine the correct version for your system by consulting the Genesys Supported Operating Environment Reference Guide. Then,

- For Solaris and Linux, download the appropriate JRE from the Oracle website.
- For Windows, download the appropriate JDK from the Oracle website.
- For AIX, download the appropriate JRE from the IBM website.

# Deploying eServices in Windows Environments

This section describes how to install the eServices components in your Windows environment.

> **Tip**
> In previous eServices releases, some components required Java Environment and Libraries for eServices and UCS (JELEU), which was supplied as part of eServices. Starting with the 8.5 release, JELEU is no longer required.

For your own deployment, be sure to use the Configuration Worksheet to help you plan the specifics of your solution.

## Deployment Steps

### Prerequisites

The following table describes the prerequisite components required before you install your eServices solution.

| Type of Component | Component Used |
|---|---|
| Operating System | A supported Windows environment. |
| Web Server | Apache Web Server |
| Web Application Container | Tomcat |
| E-mail (POP3) Server | Third-party email server |
| Java Environment | Java Runtime Environment (JRE). For supported versions, see the Genesys Supported Operating Environment Reference Guide. |
| Database Server | For supported databases, see the Genesys Supported Operating Environment Reference Guide. |
| Licence Server | License Server 9.5; License File: 7260@ESHost.eServices 8.1 also supports FLEXNet Publisher v11.9 License Manager. |
| Management Framework Components | See the Management Framework documentation for information about installing and configuring the components. |
| Stat Server | See the Stat Server documentation for installation |

| Type of Component | Component Used |
| --- | --- |
| | and configuration instructions. |
| Universal Routing Server | See the Universal Routing documentation for installation and configuration instructions. |

### Create Host

If you plan on installing eServices on the same computer as Genesys Framework, you should already have a Host object created and available for your eServices deployment. However, if you have not already done so, or if you want to install eServices on a different computer than Framework, create the `Host` object in the Configuration Layer for the machine that will host your model environment.

### Create the Databases

Create three databases in your SQL Server RDBMS. Two of these will accommodate Universal Contact Server contact information, and the other will handle Interaction Server's one table for tracking transient data. You will use the names of these databases during your creation of eServices Database Access Points (DAPs). After configuration and installation, you will run scripts against these databases to make them available to Genesys servers.

> **Important**
>
> For DB2 RDMS users, before running your database scripts, you need to create an additional table space with a corresponding buffer pool page size of at least 8 KB.

### Install and Start Interaction Workflow Samples

Install and run this component first to create various configuration objects that you will need when installing the rest of eServices.

## Post Deployment Steps

### Finishing the eServices Deployment on Windows

#### Creating Databases and Running Scripts

If you have not already done so, create the databases for Interaction Server and Universal Contact Server. For UCS, you may also want to create an Archive database to store older UCS data and ease storage in the Main UCS database.

**Running the Database Scripts**

1. At the starting directory for each component, locate the SQL script folder:

- Interaction Server—`Script`
- Universal Contact Server—`sql-scripts`

2. Run the database scripts for each component.

- UCS has two scripts. First run `ucs_drop_<dbname>.sql`; this deletes any existing tables and indexes in the UCS database. Then run `ucs_<dbname>.sql`.

> **Important**
> Only one Universal Contact Server database pair (Main and Archive) is allowed per Tenant.

- Interaction Server has a single script file.

### Verifying Connections

At this point in the eServices installation process, you must update the configuration settings for some components before continuing. See the Connections Table for a list of connections that must be set in the Configuration Layer. Refer to the eServices Options Reference for information on option settings.

1. In Configuration Manager or Genesys Administrator, verify that the required connections have been made in the eServices Application objects.
2. Add a connection from Stat Server to Interaction Server, or to Interaction Server Proxy (see "Interaction Server Proxy as Reporting Bus").
3. Add a connection from Universal Routing Server to Interaction Server.

#### Interaction Server Proxy as Reporting Bus

You can connect Stat Server to Interaction Server Proxy rather than directly to Interaction Server. This can be advantageous if you are running multiple Stat Servers. To do this,

- Deploy Interaction Server Proxy.
- Wherever this Guide speaks of connecting Stat Server to Interaction Server, instead make the connection to Interaction Server Proxy

# Testing eServices Components and the Setup

The minimum required components for testing include:

- Local Control Agent
- DB Server
- Configuration Server
- Message Server
- Solution Control Server
- Solution Control Interface
- Stat Server
- Universal Routing Server

Next you should test your installation's functionality. This requires that you implement an interaction workflow. The workflow and its associated routing strategies (Business Processes) can be very simple. The simplest way to test the setup is to use the `ABC Simple BP` and the `ABC Simple Chat BP` business processes that are installed as part of your Interaction Workflow Samples installation.

## Components

## Testing the Components

## Testing the eServices Servers

### Prerequisites

- Both Universal Contact Server and Interaction Server have connections to Stat Server (if you have not already made these connections).
- Both Stat Server and Universal Routing Server have connections to Interaction Server.
- Genesys Framework is running.
- Before testing chat and web form e-mail, restart the host(s) running the Chat Server, and ensure that

your POP3 e-mail server is up and running.

1. Start the servers in this order:
   a. DB Server for Interaction Server
   b. Interaction Server
   c. Universal Contact Server
   d. E-mail Server
   e. Chat Server
   f. Classification Server
   g. Training Server
   h. Web Server and Web Application Container for Web API Server launch
2. After each server starts, check its console window for errors.

## Testing the components using the Web-based TestTool812

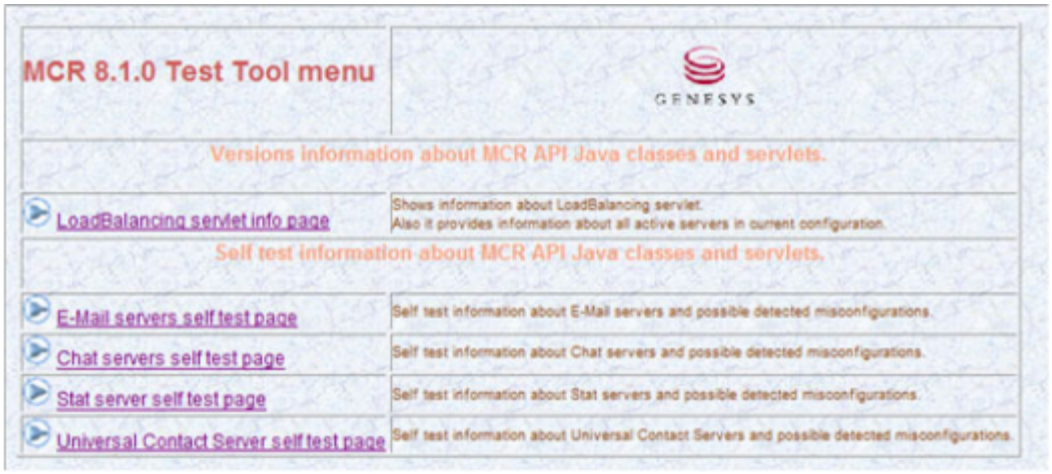
An application for testing the components, called TestTool812, is included as part of the installation of Web API Server. Use this tool to test the various Java classes and servlets that run the eServices solution.

> **Important**
>
> TestTool812 supports Internet Explorer. It does not support Mozilla Firefox.

1. Open a web browser window and enter the following URL: `http://<WebAPIServerhost>/WebAPI812/TestTool812`

   > **Important**
   >
   > The address used to access the testing application is case sensitive.

   A window similar to the following appears:




2. Click any of the links to launch the various testing tools. The results of the test display in the bottom frame of the web page.

## Testing that the desktop can handle interactions

Test agent handling of interactions by configuring and installing Workspace Desktop Edition.

## Setup

## Testing the Setup

The `ABC Simple` BP receives e-mail interactions that enter the system and delivers them to an agent group. This business process also allows the agent who works on a given interaction to reply to it and send a resulting outbound e-mail. The `ABC Simple Chat` BP does essentially the same thing, but for chat interactions; it receives chat interactions that enter the system and delivers them to an agent group. See the Universal Routing 8.1 Business Process User's Guide for more details on these and the other sample business processes. That guide also describes how to work with strategies and queues in the Interaction Design portion of IRD.

## Configuring ABC Simple BP for routing e-mail interactions

1. Set E-mail Server's `default-inbound-queue` option (in the `email-processing` section) to the value `Inbound queue`, which is the name of the inbound queue in `ABC Simple BP`.

> **Important**
>
> If you installed the Interaction Workflow Samples before using the wizard installation, you will find that two Interaction Queue objects named Inbound queue and Outbound queue already exist in your <Tenant>\Scripts folder in Configuration Manager or Genesys Administrator. Use the Inbound queue name as the value for this E-mail Server default queue option.

2. The Person object you plan to use for handling e-mail interactions must be a member of the Agent Group named E-mail distribution for processing, which serves as a target for interactions in the Process ABC strategy of this sample business process.
3. From the Interaction Design portion of IRD, if they are not already activated, activate the two strategies that this business process uses, Process ABC and Send ABC.

## Configuring ABC Simple Chat BP for routing chat interactions

Be sure you have set the following prior to testing this workflow:

1. If you installed the Interaction Workflow Samples before using the wizard installation, you will find that an Interaction Queue object named Chat inbound queue already exists in your <Tenant>\Scripts folder in Configuration Manager or Genesys Administrator. That queue is also the value for the Chat Server default option (in the endpoints:<tenant_id> section). Make sure this is the case.
2. The Person object you plan to use for handling chat interactions must be a member of the Agent Group named Chat distribution for processing, which serves as a target for interactions in the Chat inbound strategy strategy of this sample business process.
3. From the Interaction Design portion of IRD, if they are not already activated, activate the three strategies this business process uses: Chat send transcript email strategy, Chat request transcript send strategy, and Chat inbound strategy.

## Sending a test e-mail to a Desktop agent

1. If you have not already started the necessary components, do so now.
2. Open your e-mail client and send an e-mail to the POP box that E-mail Server checks (the value of that component's address option).
3. Ready an agent at the Genesys Desktop and look for the arrival of the e-mail interaction.
4. If the inbound e-mail fails (for example, E-mail Server does not see the message or E-mail Server sends the message to a bad directory), check each of the E-mail Server options, logs, and the routing strategies for possible errors.

## Starting a test chat session with a Desktop agent

1. Start the web server and the web application container, if you have not already.
2. If you have not already started the necessary components, do so now.
3. Ready an agent at the Genesys Desktop.
4. Open the Genesys MCR 8.1 Platform SDK Java Samples Pages by opening a browser navigating to `http://<WebAPIServerhost>/WebAPI812`.
5. Select the **New samples based on PSDK Java API** link to open the samples page.
6. Click `Chat Sample with "user" typing notification`.
7. Fill in customer data and click **Start Chat**. Look for the arrival of the chat interaction at the Genesys Desktop.
8. If chat does not work (for example, a disconnected from chat server message appears or the customer joins but the agent does not), check each of the manual install steps. Specifically, check the:
   - Chat Server options.
   - Web API Server options.
   - WebAPI812 parameters, contained in the `constants.jsp` file.

## Sending a test web form e-mail to a Desktop agent

1. If you have not already started the necessary component, do so now.
2. Open the Genesys MCR 8.1 Platform SDK Java Samples Pages by opening a browser navigating to `http://<WebAPIServerhost>/WebAPI812`.
3. Select the **New samples based on PSDK Java API** link to open the samples page.
4. Select the **E-mail over the Web** link.
5. Fill in customer data and click **Submit**.
6. Ready an agent at the Genesys Desktop and look for the arrival of the web form e-mail interaction.
7. If web form e-mail does not work (for example, no thank you page displays, a thank you page displays with an error, or a thank you page displays but the e-mail never pops to the desktop), check each of the manual install steps.

## Configuring WebCallback BP for routing web callback interactions

1. If you installed the Web Callback Application before using the wizard installation, you will find that an `Interaction Queue` object named `New` already exists in your `<Tenant>\Scripts` folder in Configuration Manager or Genesys Administrator. That queue is also the value for the Web API Server `wcbNewQueue` (in the `\...\SimpleSamples812\constants.jsp`). Make sure this is the case.
2. The `Person` object you plan to use for handling web callback interactions must be configured to work with voice media and be a member of the `Agent Group` named `WebCallback distribution for`

`processing`, which serves as a target for interactions in the `Delivering` strategy of this sample business process.

3. Start Knowledge Manager and import the `UCS_impex.kme` file located in the folder `<Web Callback Application installation>` to receive standard responses related to `WebCallback BP`.
4. Open the `Send Email Notification` subroutine. Select the corresponding standard responses in the two `Acknowledgement receipt` objects. Save changes in the subroutine and then in IRD.
5. From the Interaction Design portion of IRD, if they are not already activated, activate the following strategies this business process uses:
   - Preprocessing
   - Rescheduled by Agent
   - Rescheduled by Customer
   - Delivering
   - Stop By Customer
   - Stop By Agent
   - Expired Transfer Callbacks
   - Expired Conference Callbacks
   - Outbound notification email sending

## Updating the Interaction Server database for processing web callback interactions

Run the appropriate script for your Interaction Server database. The scripts are located in the following directories:

- for DB2, in the `<Web Callback Application installation>\Scripts\Db2` folder
- for MS SQL, in the `<Web Callback Application installation>\Scripts\MsSQL` folder
- for Oracle, in the `<Web Callback Application installation>\Scripts\Oracle` folder

## Sending a test web callback to a Desktop agent

1. If you have not already started the necessary components, do so now.
2. Open the Genesys MCR 8.1 Platform SDK Java Samples Pages by opening a browser and navigating to `http://<WebAPIServerhost>/WebAPI812`.
3. Select the **New samples based on PSDK Java API** link to open the samples page.
4. Select the **WebCallback** link.
5. Fill in customer data and click **Request callback**.
6. Ready an agent at the Genesys Desktop and look for the arrival of the web callback interaction.

## Building Strategies

## For More Information

See the Universal Routing 8.1 Business Process User's Guide and Universal Routing 8.1 Interaction Routing Designer Help Zip for additional instructions on creating and activating/loading routing strategies.

# Deploying eServices in UNIX Environments

This topic describes procedures for configuring and installing eServices components in UNIX environments—Solaris, AIX, and Linux—as well as for configuring the Web API components in a WebSphere or WebLogic environment. It includes the following:



> **Important**
>
> You can deploy all eServices components on UNIX platforms with the exception of the following user interfaces:
>
> - Knowledge Manager
> - Universal Contact Server Manager
> - Interaction Workflow Samples

# Deploying eServices on UNIX Hosts

Configuring and installing eServices on Solaris, AIX, or Linux follows the general procedure introduced in Model Configuration and Installation on Windows. You can use the eServices Configuration Wizards on a Windows host to configure components that will run in UNIX environments.

For the installation phase, however, you should proceed in a slightly different fashion than you would for a Windows-only environment. The main difference is that after configuration, you will need to manually copy the individual component installation packages to each UNIX host, as initially suggested in Model Configuration and Installation on Windows: "Installing eServices Components."

Also consider the guidelines under “Copying Installations to Remote Computers” in the “Genesys Wizards” section of Chapter 4, “Deployment Overview,” of the Framework 8.1 Deployment Guide (the 8.5 Framework Deployment Guide does not discuss configuration wizards).

After you copy the software to the UNIX hosts, do one of the following:

- Run the setup for individual components manually.
- Use the eServices CD on each host to run the setup for the components directly from that CD.

> **Tip**
> In previous eServices releases, some components required Java Environment and Libraries for eServices and UCS (JELEU), which was supplied as part of eServices. Starting with the 8.5 release, JELEU is no longer required.

## Procedure: Deploying eServices on Solaris, AIX, or Linux from a Windows host

**Purpose:** To suggest how you might use the eServices Configuration Wizards on a Windows host to configure and install components to other hosts for Solaris, AIX, or Linux deployments.

1. When you run the eServices Configuration Wizard, it will ask you to specify the destination for copying installation packages. Do one of the following:
   - Create a shared directory on your Solaris, AIX, or Linux computer for the installation package deployment, making sure that it is accessible from the Windows host.
   - Create a shared directory on a Windows host (as recommended in Model Configuration and Installation on Windows, in the topic: Create a Shared Directory for the Windows installation), making sure that it is accessible from the Solaris, Linux, or AIX host(s) that will run your eServices components.
     In either case, you must copy all installation packages to the Solaris, Linux, or AIX computer(s) that will host your eServices solution.
2. On a Windows machine, run the eServices Configuration Wizards as described in Model Configuration

and Installation on Windows, in the topic: Model Configuration and Installation on Windows. The Wizards copy the installation packages using a directory structure such as the following:

```
<Component A>
                IBM_AIX
                Linux
                Solaris
                Windows_2008


<Component B>
                IBM_AIX
                Solaris
                Windows_2008

...
aix
                ThirdPartyComponents


linux
                ThirdPartyComponents


solaris
                ThirdPartyComponents


windows
                CommonWizardComponentSet
                ThirdPartyComponents
```

3. After completing configuration using the Wizards, begin installation by running the installation package in `platform\ThirdPartyComponents` on each host where you plan to install components.
4. Continue installation by running the installation package for each remaining eServices component. You will find that package in the `platform>` subdirectory contained in the directory named for each component.

# Compatibility Package for Red Hat Linux

Starting with 8.5.200.02, E-mail Server supports Red Hat Enterprise Linux 8. With the Red Hat Linux subscription, after installing RHEL 8, the system may require to install the following additional compatibility libraries:

From a base RHEL 7:

- compat-db47.i686
- compat-glibc
- compat-libcap1.i686
- compat-libf2c-34.i686
- compat-libgfortran-41.i686
- compat-openldap.i686
- compat-libtiff3.i686
- libpng12.i686
- openssl098e.i686

From a base RHEL 8 repository:

- libstdc++-devel.i686
- libstdc++.i686
- libxcrypt.i686
- compat-openssl10.i686
- openssl-devel
- libnsl.i686
- zlib.i686
- glibc.i686

# Manual Installation of eServices Components

This section provides instructions on manually deploying eServices.

## Creating the Application Object

1. Create an Application object for the application if it does not already exist.
   a. Import the application template from the product CD.
   b. Create a new Application object based on the template.
2. Open the **Properties** dialog box of the Application object.
3. On the **Server Info** tab:
   - In the **Host** box, enter the name of the desired host.
   - In the **Communication Port** box, enter the port the component will use for listening.
4. On the **Start Info** tab enter some characters in the **Working Directory**, **Command Line**, and **Command Line Arguments** fields. These characters will be over-written with the correct values during the installation, but they cannot be left blank at this point.
5. On the **Connections** tab, add the appropriate connections.
6. Click **Apply**.

## Installing eServices Components on Windows

### Prerequisites

- The Application object has been configured.
- The product CD for the component you are installing.

1. Locate the `Setup.exe` for the component you are installing (available on the Product CD).
2. Double-click `Setup.exe`.
3. Enter the login information for your Configuration Server:
   - Host
   - Port
   - User

- Password

4. For components that support client-side port functionality, enable **Use Client Side Port**.
5. If you are not configuring client-side port functionality, click **Next** and continue with step 7.

> **Important**
> Client-side port configuration is not supported for every eServices component. This dialog will only appear for Chat Server and Interaction Server installations. This step is optional. Refer to the Genesys Security Deployment Guide for more information.

6. For client-side port configuration, specify the following parameters and click **Next**:
   - Port—Enter any free port number (this is not the Listening port in the **Server Info** tab of the Application object).
   - IP Address—Enter the IP Address of the computer on which you are installing and running the application.

> **Important**
> After you have entered this information, the installation process will add the necessary command-line arguments (`-transport-address` and `-transport-port`) for connecting to Configuration Server during Application startup.

7. Select the appropriate Application object from the list.
8. Click **Install**.

# Installing eServices Components on AIX, Solaris, or Linux

## Prerequisites

- The Application object has been configured.
- The product CD for the component you are installing.

1. Locate the `install.sh` file for the application you are installing.
2. Start the installation script with the command `>install.sh`
3. Press **Enter** to confirm the host name for the installation.
4. Enter the login information for your Configuration Server:
   - Host

- Port
- User
- Password

5. If the component you are installing supports client-side configuration, you will see the following prompt:
   `Do you want to use Client Side Port option (y/n)?`
   If you are not setting up client-side port configuration, enter **n** and continue to step 8.

> **Important**
>
> Client-side port configuration is not supported for every eServices component. This dialog will only appear for Chat Server and Interaction Server installations. This step is optional. Refer to the Genesys Security Deployment Guide for more information.

6. If you are setting up client-side port configuration for the initial connection to Configuration Server as described in the Genesys Security Deployment Guide, enter y.
7. Specify the following parameters:
   - Port—Enter any free port number (this is not the Listening port in the **Server Info** tab of the Application object).
   - IP Address—Enter the IP Address of the computer on which you are installing and running the application.
8. From the list of applications, select one and enter its number in the list.
9. Press **Enter** to confirm the suggested destination directory, or choose another one.
10. Answer other questions, if required.

> **Important**
>
> On some Red Hat Linux platforms, eServices components might fail to start and produce the error `./cfgutility: error while loading shared libraries: libstdc++.so.5: cannot open shared object file: No such file or directory`. Refer to Compatibility Package for Red Hat Linux for information about correcting this issue.

## Specifics

Some components require procedures beyond the general directions on this page.

- Interaction Server, including clusters.
- Universal Contact Server (UCS).

# Manual Deployment of UCS

This chapter describes procedures for manual configuration and installation of UCS and UCS Manager.

## Creating or Editing the UCS Application Object

### Prerequisites

- UCS requires one database (the main database). It can optionally use a second (archive) database to store older UCS data and ease storage in the main database.

### Procedure

1. Create the main and (optional) archive databases if you have not already done so. See also Creating Databases and Running Scripts.
2. Run the SQL script located in the `sql_scripts` folder of the UCS starting directory.
3. Create a Database Access Point (DAP) for the main database. Create a second DAP for the archive database if you have one. Use the JDBC connection type.
4. Create a UCS Application object if it does not already exist.
   a. Import the UCS application template from the Interaction Management CD.
   b. Create a new Application object based on the template.
5. Open the **Properties** dialog box of the Application object.
6. On the **Server Info** tab, enter the host name and communication port.
7. On the **Start Info** tab, enter the working directory.
8. On the **Connections** tab, add connections to:
   - Message Server
   - Stat Server
   - The two DAPs that you created in Step 3.

   Click **Apply**.
9. On the **Security** tab, in the **Log On As** area, select the **This Account** check box, and then select a Person or Access Group with privileges that are high enough to include write permission.
10. On the **Options** tab, ports section, specify for the option ucsapi a valid network port number that UCS will use for connections from Java clients.

## Creating or Editing the UCS Manager Application Object

1. Create a UCS Manager Application object if it does not exist.
   a. Import the UCS Manager application template from the Interaction Management CD.
   b. Create a new Application object based on the template.
2. On the **Connections** tab, add a connection to UCS and click **Apply**.

## Installing UCS on Windows

### Prerequisites

- The UCS Application object has been configured.
- The Interaction Management CD.

### Procedure

1. Locate the `Setup.exe` for Universal Contact Server (available on the Interaction Management CD).
2. Double-click `Setup.exe`.
3. Enter the login information for your Configuration Server:
   - Host
   - Port
   - User
   - Password
4. Select the appropriate UCS Application object from the list.
5. Click **Install**.

## Installing UCS on AIX, Solaris, or Linux

### Prerequisites

- The UCS Application object has been configured.
- The Interaction Management CD.

### Procedure

1. Locate the `install.sh` file for Universal Contact Server.
2. Start the installation script with the command `>install.sh`.
3. Provide the absolute path to the directory containing the Java executable.
4. Press **Enter** to confirm the host name for the installation.
5. Enter the login information for your Configuration Server:
   - Host
   - Port
   - User
   - Password
6. From the list of applications, select one and enter its number in the list.
7. Press **Enter** to confirm the suggested destination directory, or choose another one.
8. Answer other questions, if required.

## Installing UCS Manager

### Prerequisites

- The Universal Contact Server Manager Application object has been configured.
- The Interaction Management CD.

### Procedure

1. Locate and double-click `Setup.exe` for UCS Manager on the Interaction Management CD.
2. Click **Next**.
3. Enter or browse to the location of the destination folder. Click **Next**.
4. Select JDK version in the list. Click **Next**.
5. Click **Install**.
6. Click **Finish**.

> **Important**
>
> To run UCS Manager with Open JDK 11,

- Add to the [UCS Manager directory]/lib directory libraries:
  - jaxb-api-2.3.0.jar
  - jaxb-core-2.3.0.jar
  - jaxb-impl-2.3.0.jar
  - activation-1.1.1.jar
- Add the following to the **[UCS Manager directory]/ucsmanager.bat** file:
  - `set JAVALIB=%LIBDIR%/jaxb-api-2.3.0.jar;%LIBDIR%/jaxb-core-2.3.0.jar;%LIBDIR%/jaxb-impl-2.3.0.jar;%LIBDIR%/activation-1.1.1.jar;`
- Add the `JAVALIB` variable to `COMP_CLASSPATH`:
  - `set COMP_CLASSPATH=%UCSMANAGER%;%UNILOGIN%;%ICCLIB%;%JBCL%;%LOG4J%;%GENESYS%;%JAV`

## Configuration Server access permissions

UCS requires the following access permissions:

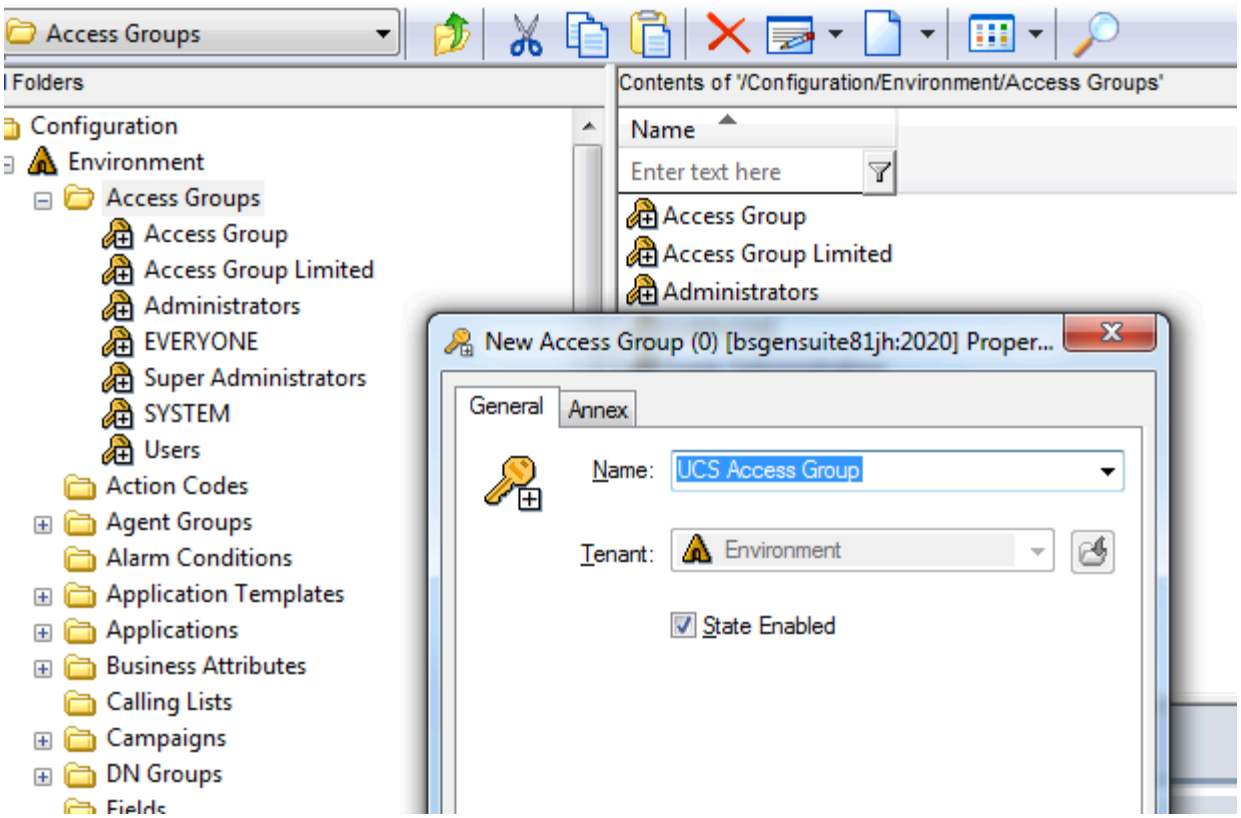
- Read permission for the Environment tenant and any other tenants configured in the UCS application.
- Full Control permission to the **Category Structure** and **Screening Rules** business attributes.

To prevent overwriting of object permissions, you must apply permissions to higher-level objects first (for example, at the tenant level) before applying permissions to lower-level objects.

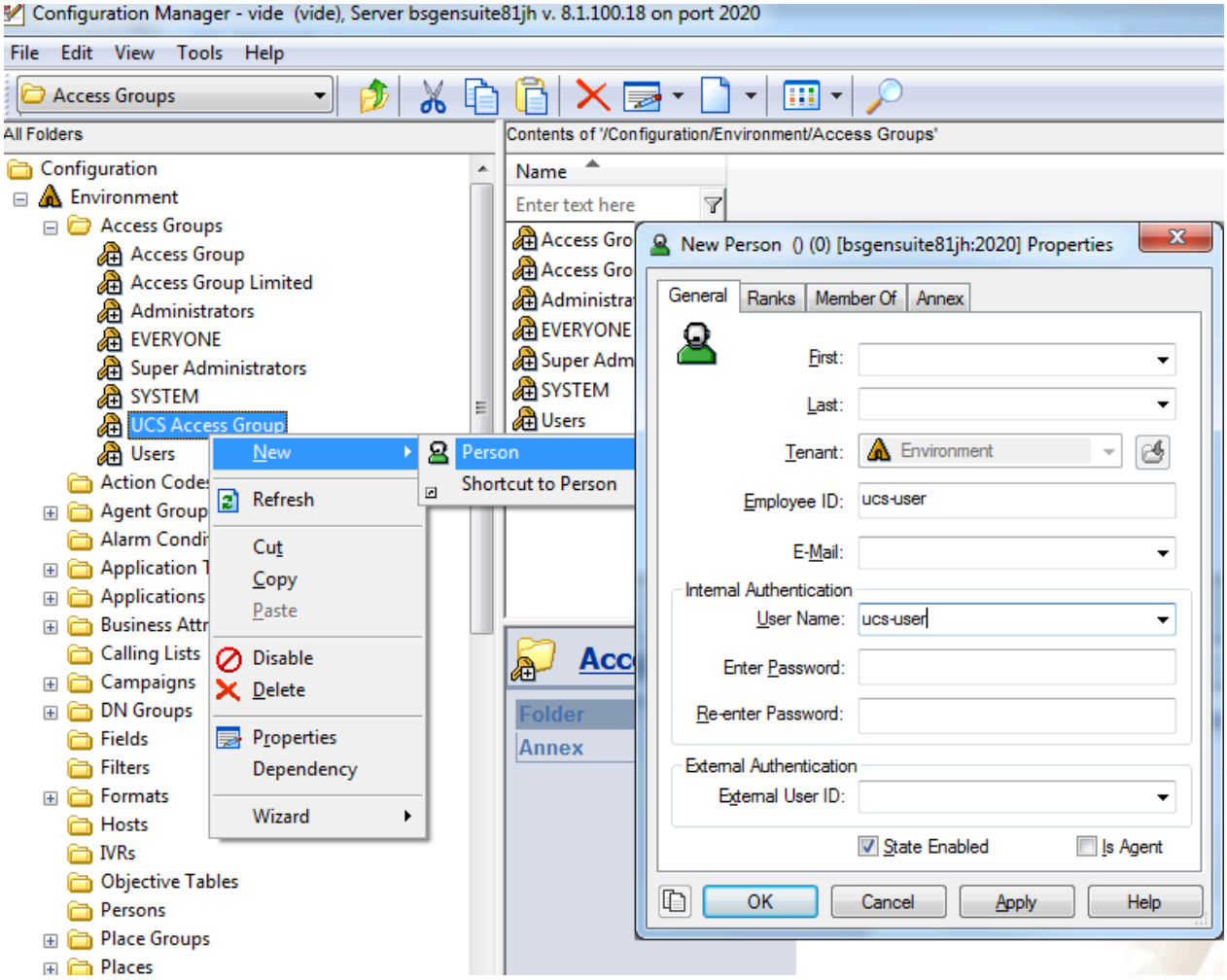
Open Configuration Manager and refer to the steps below for more information.

### Environment tenant

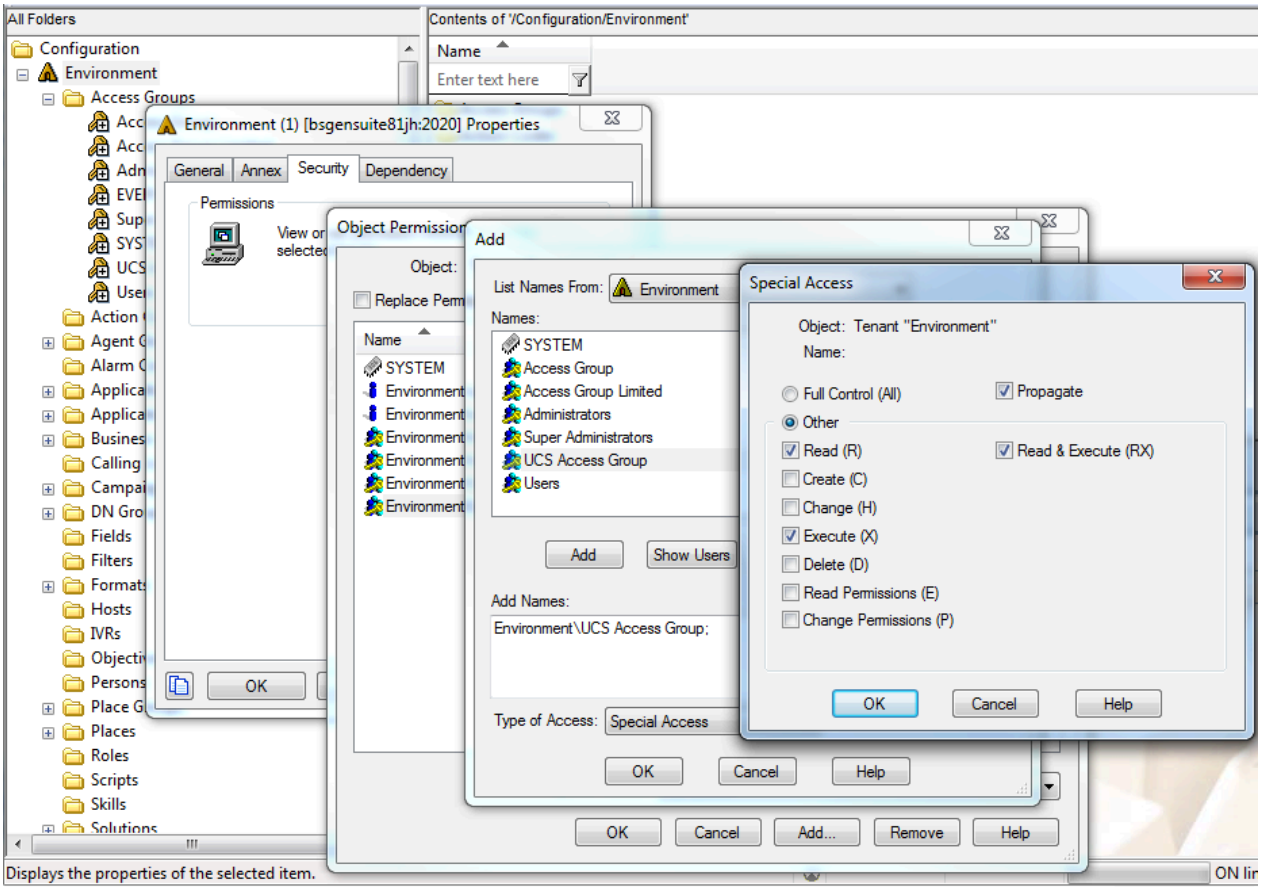
1. Create an access group.




2. Open the access group you created and create a user.

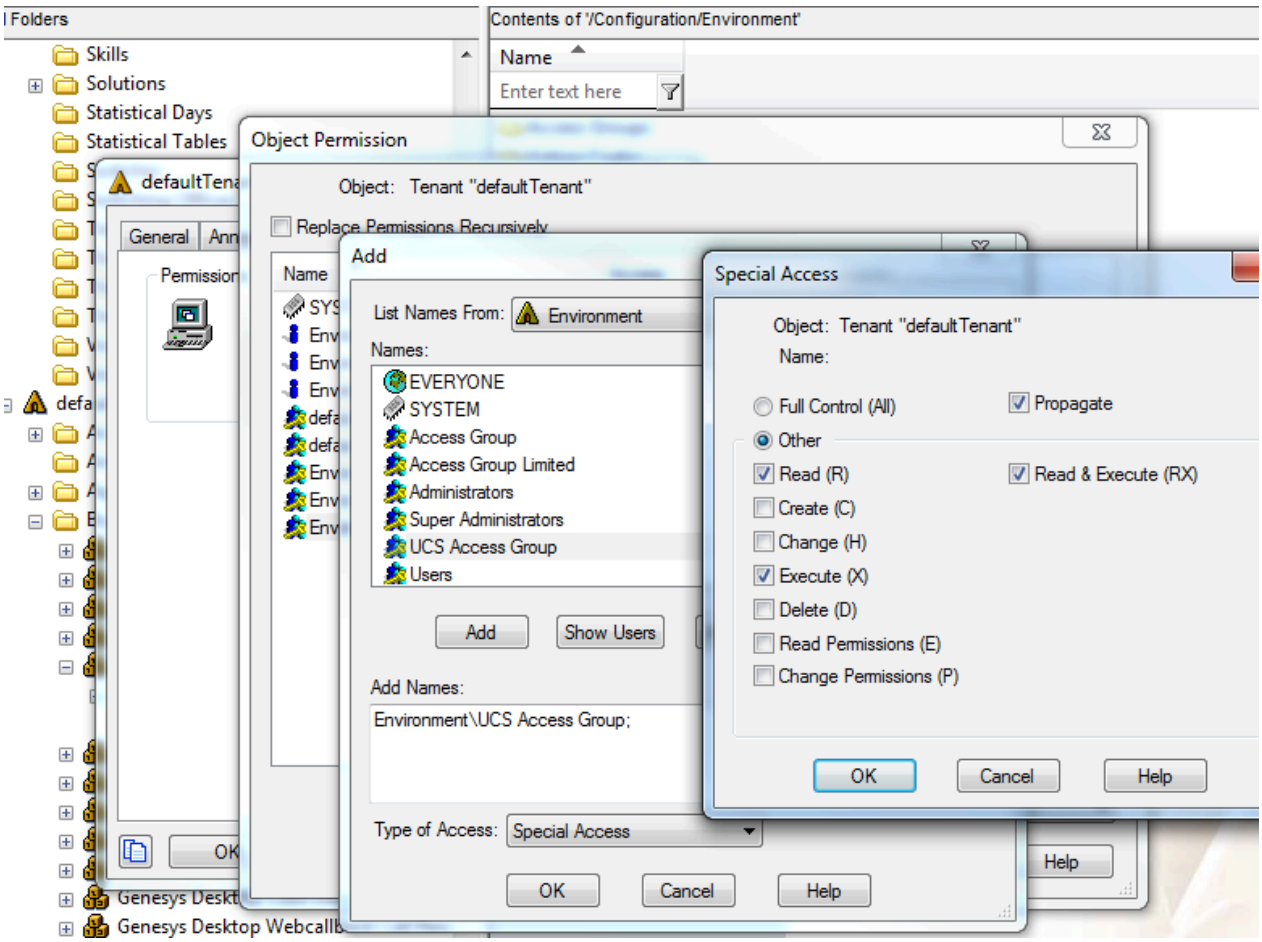



3. Assign Read and Execute (RX) permissions to the Environment tenant.

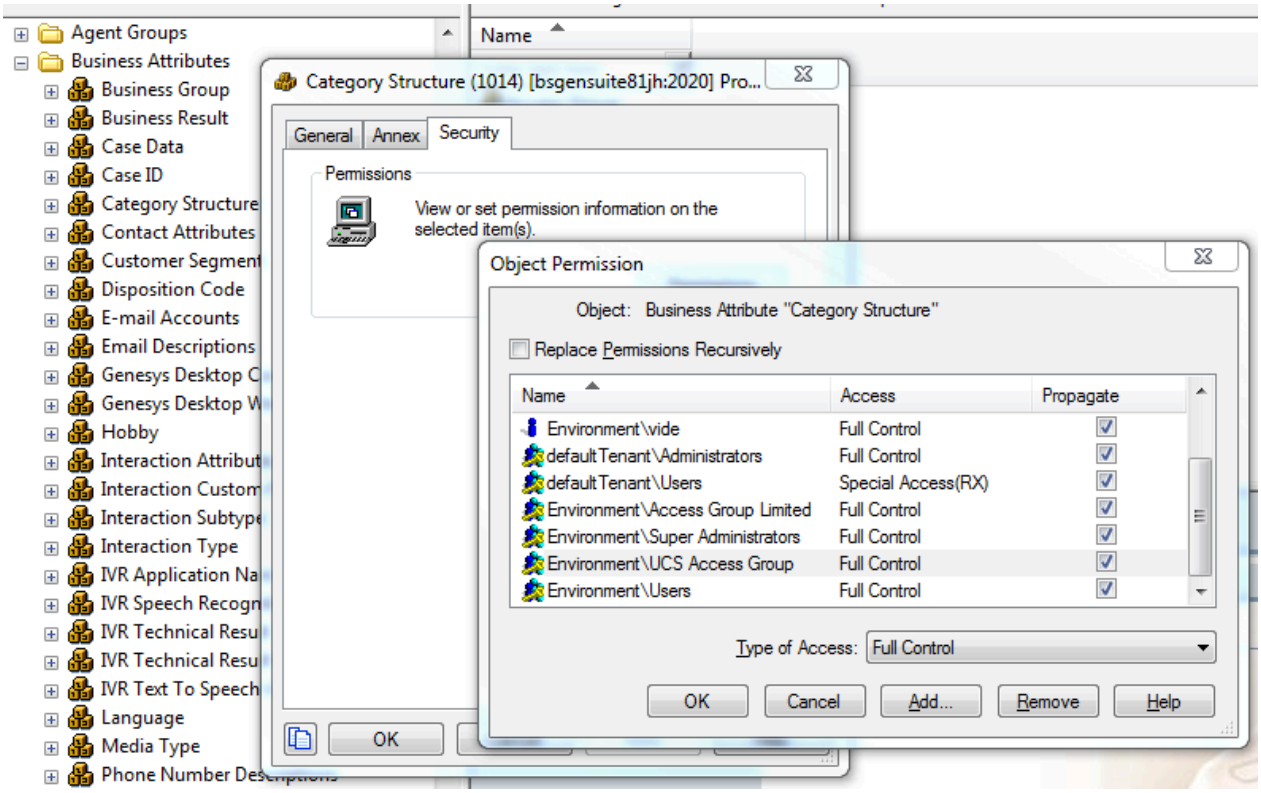



## UCS tenants

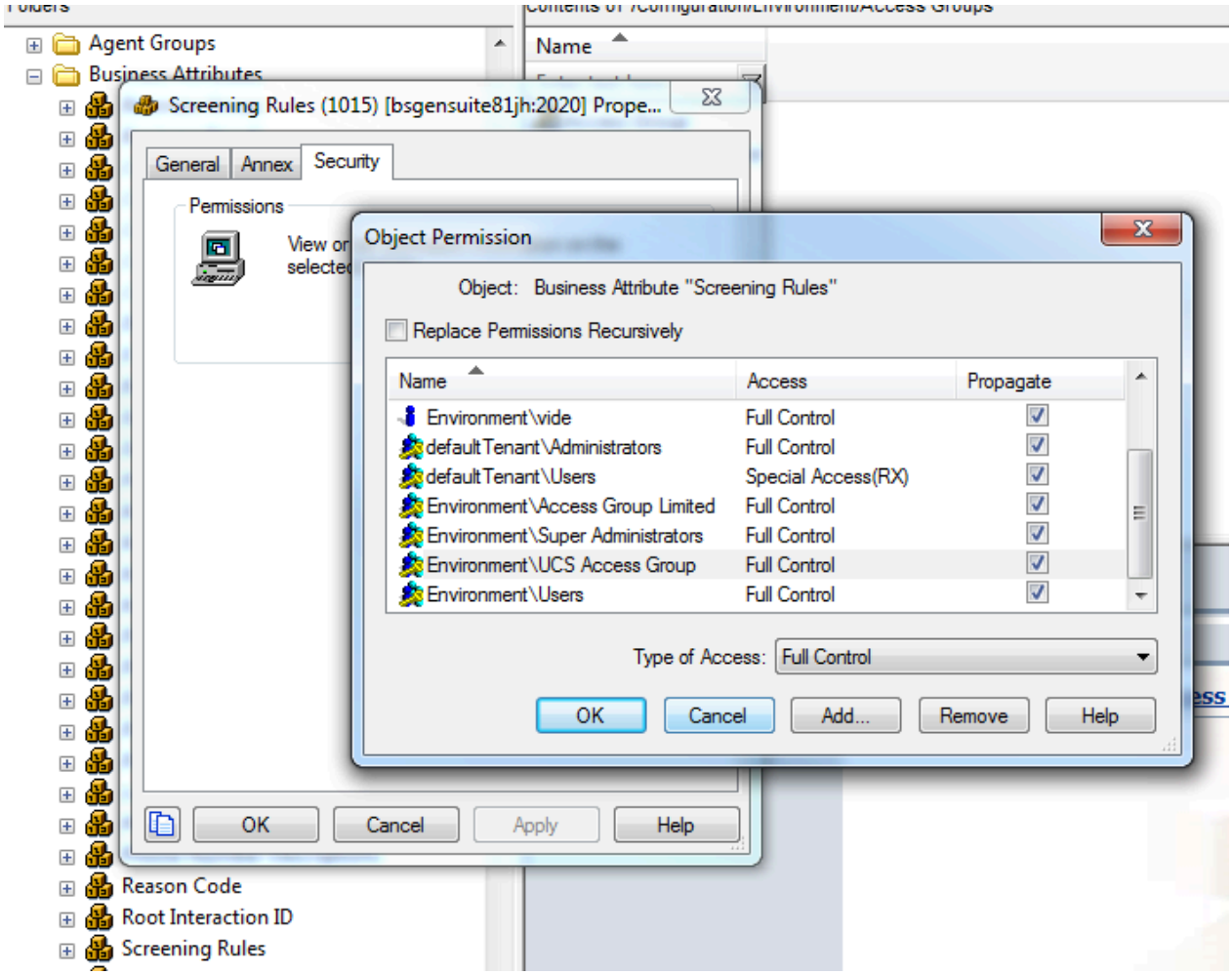
1. Assign Read and Execute (RX) permissions to the tenants configured in the UCS application.




2. Assign Full Control permission to the **Category Structure** business attribute. Ensure all **Propagate** check boxes are selected.



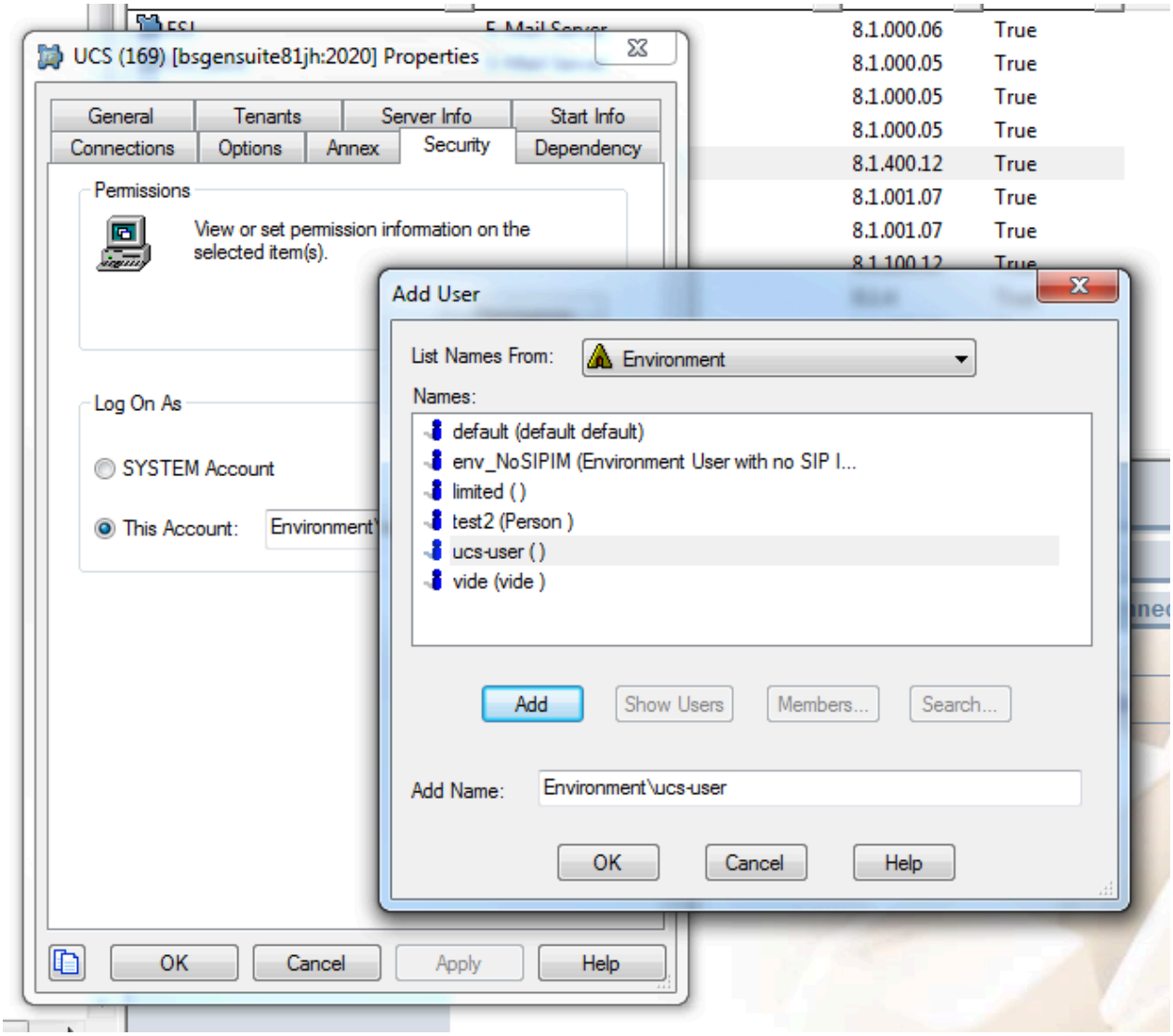

3. Assign Full Control permission to the **Screening Rules** business attribute. Ensure all **Propagate** check boxes are selected.



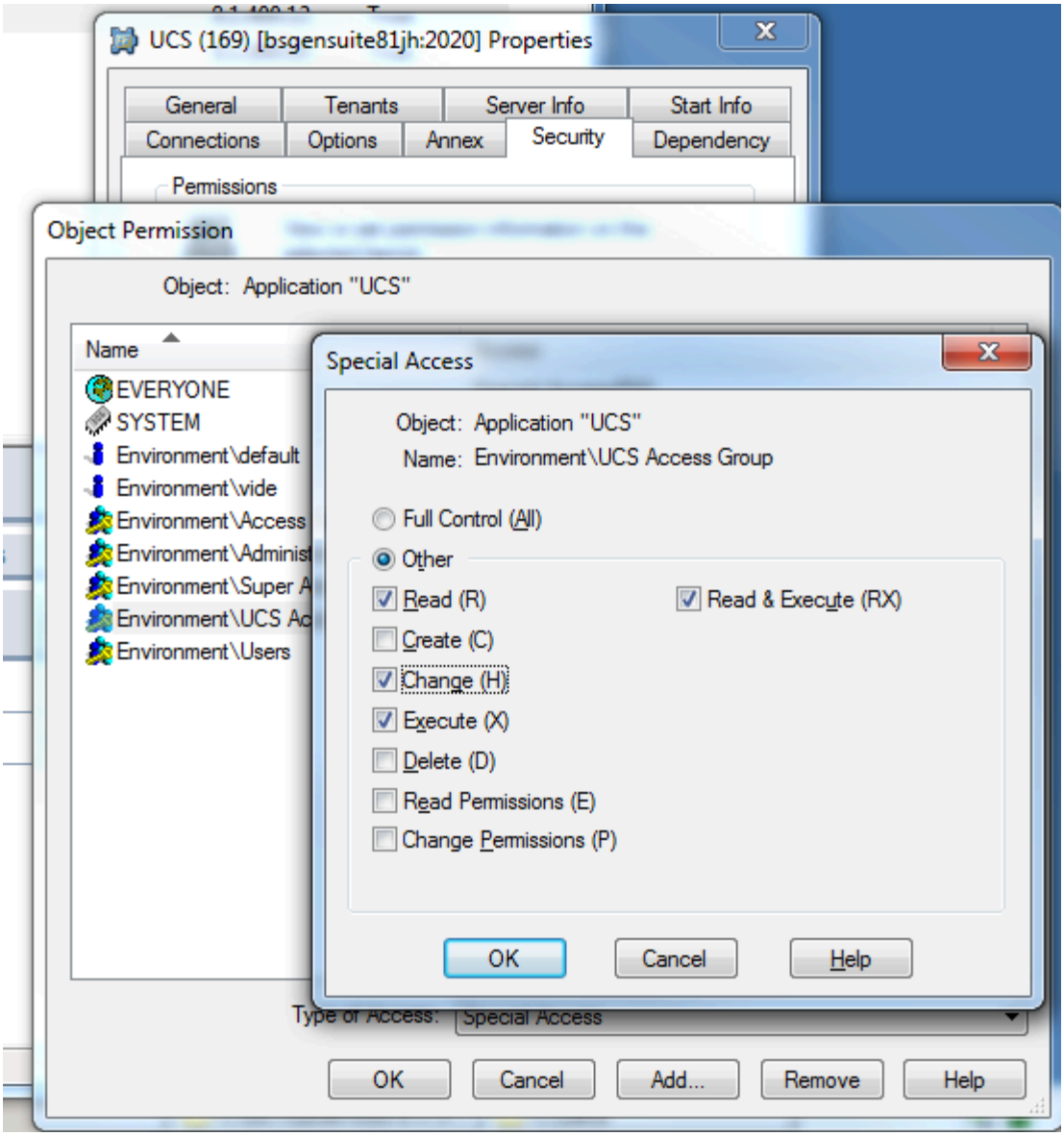

4. Repeat these steps for all tenants configured in the UCS application.

## UCS application

1. Assign the user you created in the Environment tenant section to the UCS application. Also add this user to the backup UCS application, if it exists.




2. If you intend to use UCS Manager, you must log in with the user account that has Change (write) permission for the UCS application object in Configuration Server.




## Restart UCS servers

Finally, restart the primary and backup UCS instances to apply the changes.

# Manual Deployment for UCS Proxy, Interaction Server Proxy, and SMS Server

This section describes the manual deployment of UCS Proxy, Interaction Server Proxy, and SMS Server.

## Introduction

Large numbers of custom desktop (ESP client) connections to Interaction Server and UCS may give rise to performance issues.

### Important

- UCS Proxy and Interaction Server Proxy support High Availability in Warm Standby mode.
- UCS Proxy and Interaction Server Proxy are *not* supported with Genesys Agent Desktop and desktops that connect using Interaction SDK. A custom desktop that connects using Platform SDK Contact can use UCS Proxy and Interaction Server Proxy.

To mitigate the issues caused by a high load on the server, Genesys introduced Interaction Server Proxy and UCS Proxy in release 7.6.1. Desktop applications can be configured to connect to these Proxy servers instead of the main server, significantly reducing the load on the server. For example, it is easier for the server to handle 20,000 clients that operate through ten proxies (only ten connections) than to handle the same 20,000 clients that each connect separately.

### Important

- These Proxy components are not part of the eServices Configuration Wizards and must be installed manually.
- UCS Proxy only supports connections from desktop applications; Interaction Server Proxy supports connections from desktop applications and from Stat Server.

Because of the many variables in deployment (choice of operating system, number of clients, details of architecture) is so broad, it is not possible to provide exact guidelines as to when deploying Proxy servers would be advantageous. However it may be stated that you can anticipate performance issues when the number of clients exceeds 10,000.

### Requirements

These components work with Universal Contact Server and Interaction Server. In order for the respective Proxy servers to work properly, all components that are needed by Universal Contact Server and Interaction Server must be installed. This will depend on your environment and how you are using Universal Contact Server and Interaction Server. For more details, refer to the chapter that is applicable to you:

- Deploying eServices on Windows for a typical eServices Solution running on Windows.
- Deploying eServices on UNIX for a sample deployment of an eServices Solution running on UNIX.
- Manual Deployment of eServices Components, for environments that are not using all eServices components, but that do require Universal Contact Server, such as Voice Callback.

## Deploying UCS Proxy, Interaction Server Proxy and SMS Server

This section describes how to manually install UCS Proxy, Interaction Server Proxy and SMS Server.

## UCS Proxy

## UCS Proxy Deployment

## Creating the Application Object

### Prerequisites

- The Interaction Management CD.

1. Create an Application object for the Proxy if it does not already exist.
   a. Import the correct application template from the Interaction Management CD for the UCS Proxy.
   b. Create a new Application object based on the template.
3. Open the **Properties** dialog box of the Application object.
4. On the **Server Info** tab:
   - In the **Host** text box, enter the name of the desired host.
   - In the **Communication Port** text box, enter the port the UCS Proxy will use for listening.

5. On the **Start Info** tab enter some characters in the **Working Directory**, **Command Line**, and **Command Line Arguments** fields. These characters will be over-written with the correct values during the installation, but they cannot be left blank at this point.
6. On the **Connections** tab, add connections to:
   - The primary Universal Contact Server (for UCS Proxy). This connection is mandatory.
   - Message Server (optional)
7. Click **Apply**.
8. If you would like to configure your UCS Proxy for HA, repeat this procedure for the second instance.

## Installing UCS Proxy

### Prerequisites

- The UCS Proxy Application object has been configured.
- The Interaction Management CD.

1. Locate the `Setup.exe` for UCS Proxy (available on the Interaction Management CD).
2. Double-click `Setup.exe`.
3. Enter the login information for your Configuration Server:
   - Host
   - Port
   - User
   - Password
4. Select the appropriate UCS Proxy Application object from the list.
5. Click **Install**.

## Installing UCS Proxy on AIX, Solaris, or Linux

### Prerequisites

- The UCS Proxy Application object has been configured.
- The Interaction Management CD.
- Review the section Compatibility Package for Red Hat Linux.

1. Locate the `install.sh` file for UCS Proxy.
2. Start the installation script with the command `install.sh`.
3. Press **Enter** to confirm the host name for the installation.

4. Enter the login information for your Configuration Server:
   - Host
   - Port
   - User
   - Password
5. From the list of applications, select one and enter its number in the list.
6. Press **Enter** to confirm the suggested destination directory, or choose a desired one.
7. Answer other questions, if required.

> **Important**
>
> On some Red Hat Linux platforms, eServices components might fail to start and produce the error `./cfgutility: error while loading shared libraries: libstdc++.so.5: cannot open shared object file: No such file or directory`. Refer to Compatibility Package for Red Hat Linux for information about correcting this issue.

## Configuring the Desktop Application to Use UCS Proxy

> **Important**
>
> UCS Proxy is not supported with Genesys Agent Desktop and desktops that connect using Interaction SDK. A custom desktop that connects using Platform SDK Contact can use the Proxy server.

### Prerequisites

- Your desktop application is installed and configured.
- The UCS Proxy is installed and configured.

1. Login to Configuration Manager or Genesys Administrator.
2. Locate the desktop Application object and open its **Properties** dialog box.
3. On the **Connections** tab, remove the connection to the main Universal Contact Server.
4. Add a connection to the UCS Proxy.
5. Click **Apply**

# Interaction Server Proxy

# Interaction Server Proxy Deployment

## Creating the Application Object

### Prerequisites

- The Interaction Management CD.

### Procedure

1. Create an Application object for the Proxy if it does not already exist.
    a. Import the correct application template from the Interaction Management CD for the Interaction Server Proxy.
    b. Create a new Application object based on the template.
3. Open the **Properties** dialog box of the Application object.
4. On the **Server Info** tab:
    - In the **Host** text box, enter the name of the desired host.
    - In the **Communication Port** text box, enter the port the Interaction Server Proxy will use for listening.
5. On the **Start Info** tab enter some characters in the **Working Directory**, **Command Line**, and **Command Line Arguments** fields. These characters will be over-written with the correct values during the installation, but they cannot be left blank at this point.
6. On the **Connections** tab, add connections to:
    - The primary Interaction Server. This connection is mandatory.
    - E-mail Server. This connection is required in order to send outbound e-mails from Workspace Desktop Edition.
    - Message Server (optional)
7. Click **Apply**.
8. If you would like to configure your Interaction Server Proxy for HA, repeat this procedure for the second instance.

## Installing Interaction Server Proxy

### Prerequisites

- The Interaction ServerProxy Application object has been configured.
- The Interaction Management CD.

### Procedure

1. Locate the `Setup.exe` for Interaction Server Proxy (available on the Interaction Management CD).
2. Double-click `Setup.exe`.
3. Enter the login information for your Configuration Server:
   - Host
   - Port
   - User
   - Password
4. Select the appropriate Interaction Server Proxy Application object from the list.
5. Click **Install**.

## Installing Interaction Server Proxy on AIX, Solaris, or Linux

### Prerequisites

- The Interaction Server Proxy Application object has been configured.
- The Interaction Management CD.
- Review the section Compatibility Package for Red Hat Linux.

### Procedure

1. Locate the `install.sh` file for Interaction Server Proxy.
2. Start the installation script with the command `install.sh`.
3. Press **Enter** to confirm the host name for the installation.
4. Enter the login information for your Configuration Server:
   - Host
   - Port
   - User
   - Password
5. From the list of applications, select one and enter its number in the list.
6. Press **Enter** to confirm the suggested destination directory, or choose a desired one.

7. Answer other questions, if required.

> **Important**
>
> On some Red Hat Linux platforms, eServices 8.1 components might fail to start and produce the error `./cfgutility: error while loading shared libraries: libstdc++.so.5: cannot open shared object file: No such file or directory`. Refer to Compatibility Package for Red Hat Linux for information about correcting this issue.

## Configuring the Desktop Application to Use Interaction Server Proxy

> **Important**
>
> Interaction Server Proxy is not supported with Genesys Agent Desktop and desktops that connect using Interaction SDK. A custom desktop that connects using Platform SDK Contact can use the Proxy server.

### Prerequisites

- Your desktop application is installed and configured.
- The Interaction Server Proxy is installed and configured.

### Procedure

1. Login to Configuration Manager or Genesys Administrator.
2. Locate the desktop Application object and open its **Properties** dialog box.
3. On the **Connections** tab, remove the connection to the main Interaction Server.
4. Add a connection to the Interaction Server Proxy.
5. Click **Apply**

## SMS Server

# SMS Server Deployment

## Creating the Application Object

### Prerequisites

- The SMS Server CD.
- The following Java versions:
  - In release 8.1.300.14 and later,
    - Windows: JDK 1.7
    - UNIX: JRE 1.7
  - In release 8.1.3 prior to 8.1.300.14,
    - Windows: JDK 1.6 or JDK 1.7
    - UNIX: JRE 1.6 or JRE 1.7
  - In releases prior to 8.1.3, JDK 1.6 on Windows and JRE 1.6 on UNIX

1. Create an Application object for the SMS Server if it does not already exist.
   a. Import the correct application template from the SMS Server CD.
   b. Create a new Application object based on the template.
3. Open the **Properties** dialog box of the Application object.
4. On the **Server Info** tab:
   - In the **Host** text box, enter the name of the desired host.
   - In the **Communication Port** text box, enter the port the SMS Server will use for listening.
5. On the **Start Info** tab enter some characters in the **Working Directory**, **Command Line**, and **Command Line Arguments** fields. These characters will be over-written with the correct values during the installation, but they cannot be left blank at this point.
6. On the **Connections** tab, add connections to:
   - The primary Interaction Server. This connection is mandatory.
   - Message Server (optional)
7. If this is for a multi-tenant environment, add the tenant(s) on the **Tenants** tab.
8. Click **Apply**.

## Installing SMS Server on Windows

### Prerequisites

- The SMS Server Application object has been configured.
- Java is installed on the host.
- The SMS Server CD.

1. Locate the `Setup.exe` for SMS Server (available on the SMS Server CD).
2. Double-click `Setup.exe`.
3. Select Java.
4. Select **Use Client Side Port** if applicable.
5. Enter the login information for your Configuration Server:
   - Host
   - Port
   - User
   - Password
6. Select the appropriate SMS Server Application object from the list.
7. Select the destination location.
8. Select JDK.
9. Click **Install**.

> **Important**
>
> Refer to the SMS Server Options Reference for additional information. You must properly configure the x-jsms-config-file option before using SMS Server.

## Installing SMS Server on AIX, Solaris, or Linux

### Prerequisites

- The SMS Server Application object has been configured.
- JRE x64, or x32 (depending on Operating System version) is installed.

> **Important**

- For Oracle, Java version 1.6.0_26 or higher (within the 1.6 family) is supported. In release 8.1.3 and higher, JRE 1.7 is also supported.
- For IBM, Java (OS AIX) version 1.6.0 build pap6460sr9fp2-20110627_03(SR9 FP2) for AIX or higher (within the 1.6 family) is supported. In release 8.1.3 and higher, Java 1.7 is also supported.

- The SMS Server CD.
- Review the section Compatibility Package for Red Hat Linux.

1. Locate the `install.sh` file for SMS Server.
2. Start the installation script with the command `install.sh`.
3. Press **Enter** to confirm the host name for the installation.
4. Enter the login information for your Configuration Server:
   - Host
   - Port
   - User
   - Password
5. Choose whether the application will use Client Side Port.
6. From the list of applications, select one and enter its number in the list.
7. Press **Enter** to confirm the suggested destination directory, or choose another one inside the directory referred to by the `GES_HOME_810` variable.
8. Answer other questions, if required.

### Important

On some Red Hat Linux platforms, eServices 8.1 components might fail to start and produce the error `./cfgutility: error while loading shared libraries: libstdc++.so.5: cannot open shared object file: No such file or directory`. Refer to Compatibility Package for Red Hat Linux for information about correcting this issue.

### Important

> Refer to the SMS Server Options Reference for additional information. You must properly configure the x-jsms-config-file option before using SMS Server. This option is required for MMS processing.

## Configuring SMS Server for Chat Server (Optional)

SMS Server will work with Chat Server in session mode:

1. Add a connection to Solution Control Server.
2. Add a connection to one or more Chat Servers.

## Configuring eServices Components for SMS Server

### Prerequisites

- The SMS Server Application object exists.

1. In Configuration Manager or Genesys Administrator, locate the Interaction Server Application object and open its properties.
2. On the **Connections** tab, add a connection to SMS Server.
3. Click **Apply**.

# Wizards

## Using the eServices Configuration Wizards

For information about installing the eServices components using the wizards, see Using the eServices Configuration Wizards.

# Deploying Java Components on Windows and UNIX

## Windows

1. Verify that the Oracle JDK for Windows x64 of the correct version is installed. The correct versions are:
   - Version 1.6.0_26 or higher within the 1.6 family
   - Version 1.7
2. Run the component's IP (located in the Windows_x64 folder on the CD).
3. When you are prompted with a list of JDK choices, select the required version (1.6.0_26 or higher) of Oracle JDK.

## UNIX

1. Verify that JRE x64 for UNIX is installed.
   - For Oracle JRE, the version must be one of the following:
     - 1.6.0_26 or higher, within the 1.6 family
     - 1.7
   - For IBM JRE (for AIX), the version must be one of:
     - 1.6.0 build pap6460sr9fp2-20110627_03(SR9 FP2) or higher, within the 1.6 family
     - 1.7
2. Run the component's UNIX IP. When prompted, enter the path to the Java executable (from JRE x64). Enter the path to the Java executable (from JREx64) when you are prompted to do so during the installation.

# Transport Layer Security

eServices 8.5 supports Transport Layer Security (TLS) protocol to secure data exchange between components. The following tables list various component connections and indicates whether the specific connection supports TLS. In these tables, the components listed in the first column are connecting to the components that are listed in the top row.

Further information on using TLS is available as follows:

- Refer to the Genesys Security Deployment Guide for information about how to configure TLS for eServices components.
- For detailed information on using TLS with UCS and UCS clients, see the "TLS" section of the Security and Authentication page of the Context Services User's Guide.
- UCS, E-mail Server, and UCS Manager use Genesys Platform SDK to implement TLS connectivity. Refer to the Using and Configuring Security Providers page of the Platform SDK Developer's Guide for detailed configuration of TLS providers, as well as recommended Java versions for TLS on Windows.

> **Important**
>
> When configuring secured connections for Interaction Server on Unix platforms, you must
>
> - Use Genesys Security Pack on UNIX 8.5.x.x or later.
> - Set the environment variable `GCTI_SECPACK_USE_MTCB` to 1.

## TLS Support 1

## TLS Support 1 Connections

| Connection | Config Server | Config Server Proxy | Solution Control Server | Message Server | Interaction Server | Universal Contact Server |
|---|---|---|---|---|---|---|
| **Interaction Server** | Yes | Yes | N/A | Yes | Yes | Yes |
| **Interaction Server** | Yes | Yes | N/A | Yes | Yes | N/A |

| Connection | Config Server | Config Server Proxy | Solution Control Server | Message Server | Interaction Server | Universal Contact Server |
|---|---|---|---|---|---|---|
| **Proxy** | | | | | | |
| **Universal Contact Server** | Yes | Yes | N/A | Yes | N/A | |
| **Universal Contact Server Proxy** | Yes | Yes | N/A | Yes | N/A | Yes |
| **E-mail Server** | Yes | Yes | N/A | Yes | Yes | Yes |
| **Chat Server** | Yes | Yes | N/A | Yes | Yes | Yes |
| **SMS Server** | Yes | Yes | Yes | Yes | Yes | N/A |
| **Social Messaging Server** | Yes | Yes | N/A | Yes | Yes | Yes |
| **Classification Server** | Yes | Yes | N/A | No | N/A | No |
| **Web API Java** | Yes | Yes | Yes | No | Yes | Yes |
| **Web API .NET** | Yes | Yes | Yes | No | Yes | Yes |

## TLS Support 2

## TLS Support 2 Connections

## TLS Support 2

| Connection | E-mail Server | Chat Server | SMS Server | Social Messaging Server | Classification Server |
|---|---|---|---|---|---|
| **Interaction Server** | Yes | Yes | Yes | Yes | No |
| **Universal Contact Server** | N/A | N/A | N/A | N/A | N/A |
| **Universal** | N/A | N/A | N/A | N/A | N/A |

| Connection | E-mail Server | Chat Server | SMS Server | Social Messaging Server | Classification Server |
|---|---|---|---|---|---|
| **Contact Server Proxy** | | | | | |
| **E-mail Server** | | N/A | N/A | N/A | N/A |
| **Chat Server** | N/A | | N/A | N/A | N/A |
| **SMS Server** | N/A | Yes | | N/A | N/A |
| **Social Messaging Server** | N/A | Yes | N/A | | N/A |
| **Classification Server** | N/A | N/A | N/A | N/A | |
| **Web API Java** | Yes | Yes | N/A | N/A | N/A |
| **Web API .NET** | Yes | Yes | N/A | N/A | N/A |

## TLS Support 3

## TLS Support 3 Connections

| Connection | Web API Java | Web API .NET | DB Server | Universal Routing Server | Stat Server |
|---|---|---|---|---|---|
| **Interaction Server** | N/A | N/A | Yes | N/A | Yes |
| **Universal Contact Server** | N/A | N/A | N/A | N/A. (See Configuring TLS for Oracle Database connections from UCS release 8.5.300.11. | No |
| **Universal Contact Server Proxy** | N/A | N/A | N/A | N/A | N/A |
| **E-mail Server** | N/A | N/A | N/A | N/A | N/A |
| **Chat Server** | N/A | N/A | N/A | N/A | N/A |
| **SMS Server** | N/A | N/A | N/A | N/A | N/A |
| **Social** | N/A | N/A | N/A | N/A | N/A |

| Connection | Web API Java | Web API .NET | DB Server | Universal Routing Server | Stat Server |
|---|---|---|---|---|---|
| **Messaging Server** | | | | | |
| **Classification Server** | N/A | N/A | N/A | N/A | N/A |
| **Web API Java** | | | N/A | N/A | Yes |
| **Web API .NET** | | | N/A | N/A | Yes |

> **Important**
>
> Training Server does not support the TLS connection type. Stat Server and Universal Routing Server support TLS with their connections to Interaction Server. Universal Contact Server Proxy supports TLS with its connection to Universal Contact Server. No other connections to eServices components are supported by UCS Proxy.

# Configuring TLS for Oracle database connections (8.5.300.12+)

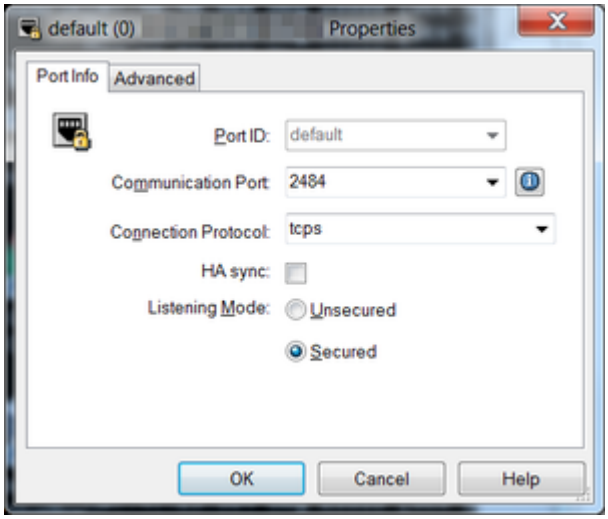
## TLS configuration for UCS connecting to an Oracle database

Note that mutual TLS is not supported, only simple TLS.

## Configuring UCS DAP in Configuration Server

The below procedure applies to both main and archive DAPs.

1. Stop the UCS server.
2. Open the Oracle DAP used by UCS.
3. Select the **Server Info** tab.
4. Open the port to be secured.
5. Type `tcps` in the **Connection Protocol** textbox.
6. Select Listening Mode **Secured**.
7. Click **OK.**



## UCS Configuration

1. Retrieve the required certificates.
   Ask the Oracle administrator to provide the Oracle certificate chain. This may include the root CA if your certificates are self-signed.

2. Register the Oracle certificate(s) in the **truststore** file.
   Register on the UCS 8.5 server hosts (**primary** and **backup**) the chain of certificates that will be used to validate the Oracle server certificate with the Java **truststore** file. The following example uses keytool and a self-signed CA:

   ```
   keytool -import -trustcacerts -keystore /usr/lib/jvm/java-8-oracle/jre/lib/
   security/cacerts -storepass changeit -noprompt -alias "Genesys Internal CA" -file
   ~/ca-intermediate.crt
   ```

3. Configure UCS to use your **truststore** file.
   Truststore can be configured by specifying Java environment variables in **contactServer.sh** or **ContactServerDriver.ini**.

   ```
   -Djavax.net.ssl.trustStore=/usr/lib/jvm/java-8-oracle/jre/lib/security/cacerts

   -Djavax.net.ssl.trustStoreType=jks

   -Djavax.net.ssl.trustStorePassword=changeit
   ```

   You may also need to specify common cipher suites to use. Please refer to Java SSL vendor documentation for a more details.

4. Start UCS 8.5 server.

# Advance Secure Connection Configuration for PostGreSQL

Starting with release 8.5.300.32, UCS supports advanced secure configuration connection for PostGreSQL, including FIPS. This configuration can be achieved in any of three ways:

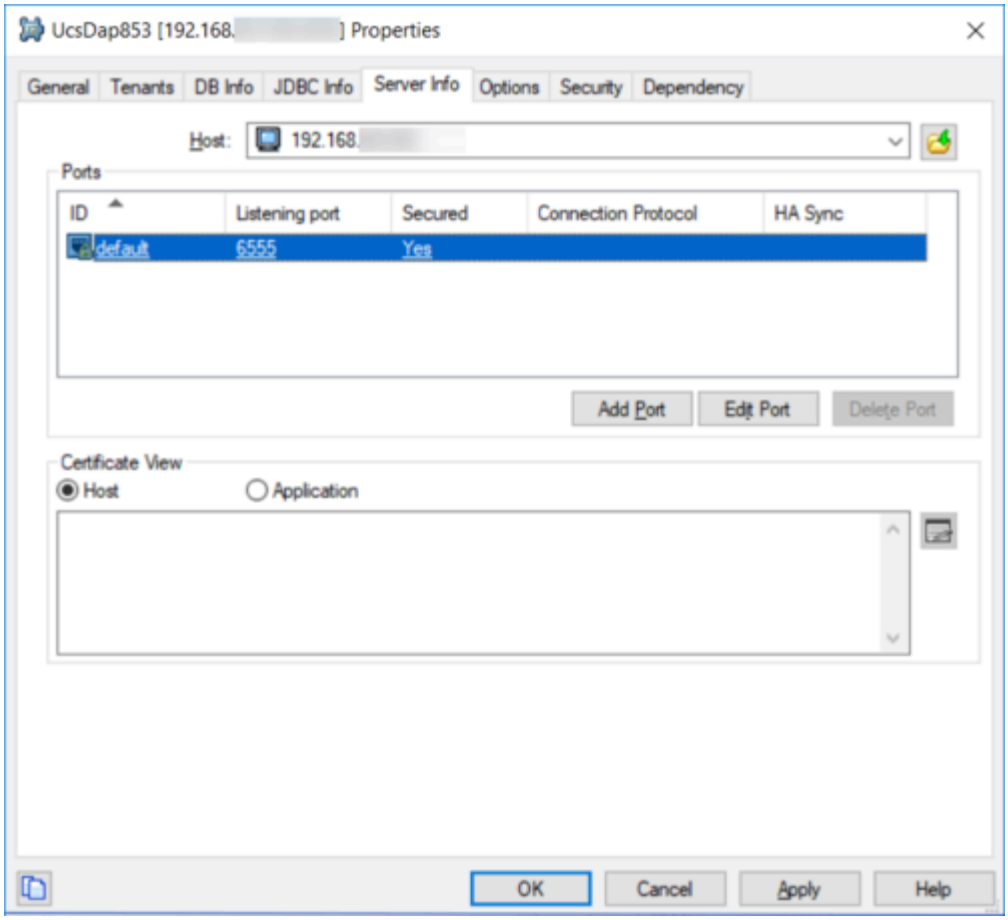
- Within the DAP application itself (two methods are available)
- On the connection link between the UCS application in Configuration Server and the Postgres DAP application
- On the host object where the Postgres database resides

Configuration approaches are described below.

## Configuration within the DAP Application

### Method 1

1. Enable the secure port option:

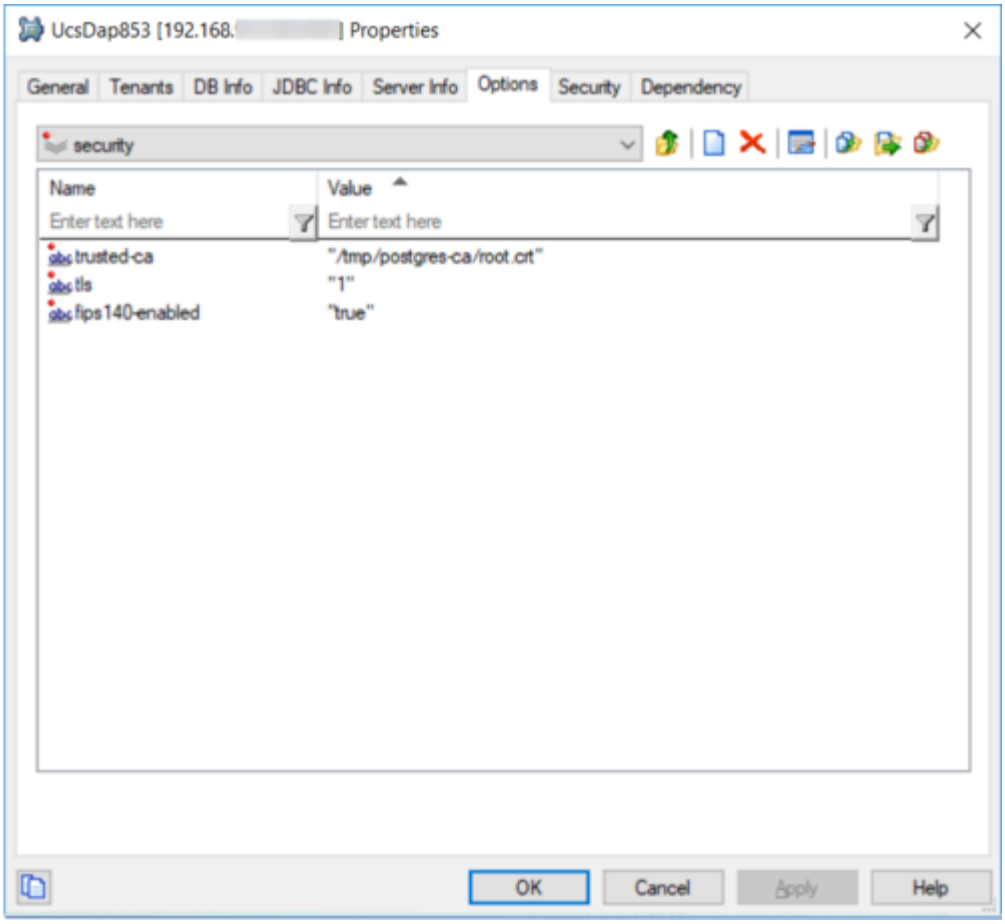



2. On the **Options** tab, create a section called **[security]**. This is because the connection is client only.
3. In the new **[security]** section:
   a. Add all TLS-related information, such as certificates information.
   b. Manually add the option **fips140-enabled** and set its value to `true` to enable FIPS mode on the database connection. Note: FIPS mode is never enabled by default and must be explicitly (manually) set.
   c. For Postgres, if the Postgres **sslmode** is set to `verify-ca` or `verify-full`, create a CA root certificate.

### Example 1:

```
[security]
tls=1
fips140-enabled=true
trusted-ca=<certificate path>
```

### Example 2:




All parameters from Genesys TLS configuration are available but do not necessarily apply to the Postgres TLS connection. Specifically, the following options are not applicable to the Postgres database connection:

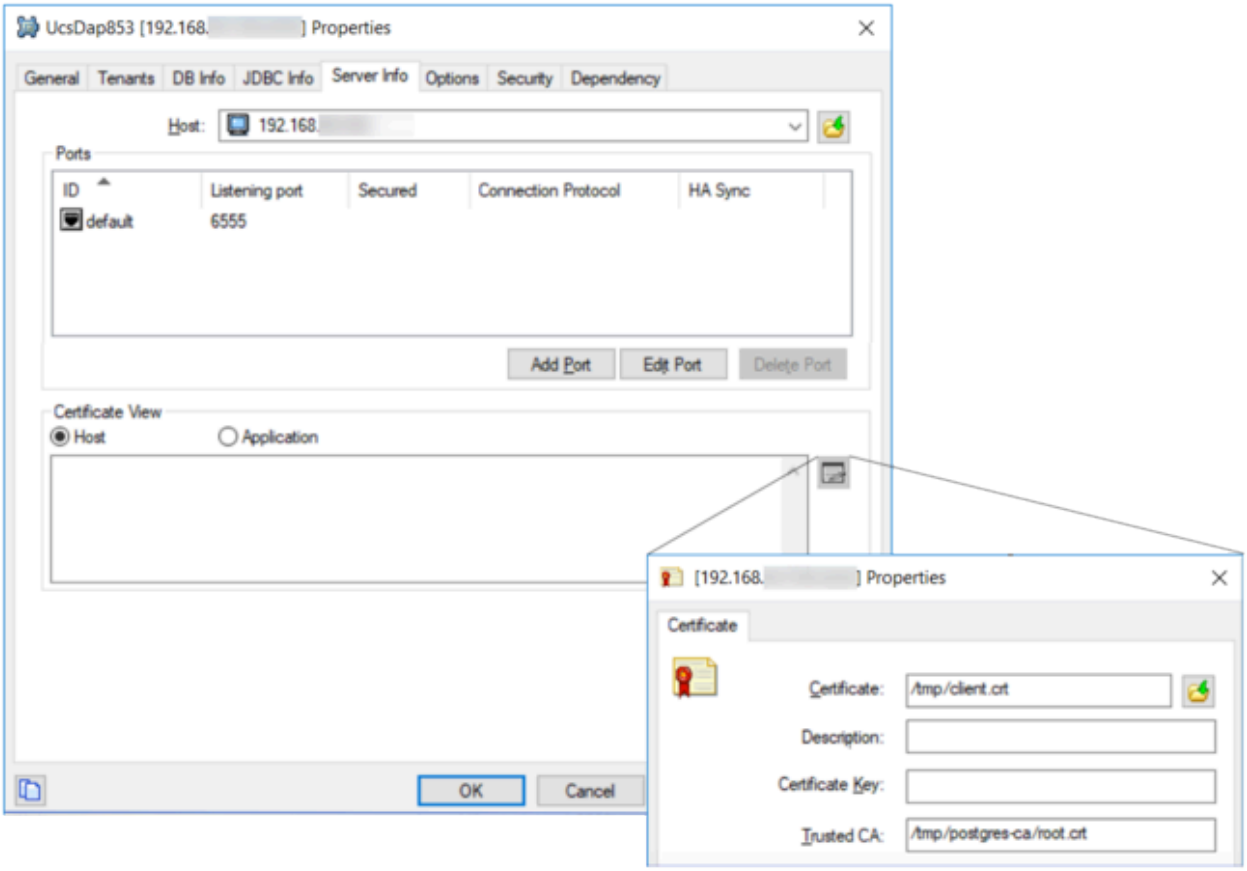
- **provider** (hard-coded to Bouncy Castle)
- **tls-mutual** (Postgres server-side configuration)
- **tls-crl** (Postgres server-side configuration)
- **cipher-list** (Postgres server-side configuration)
- **sec-protocol** (Postgres server-side configuration)
- **tls-version** (Postgres server-side configuration)
- **protocol-list** (Postgres server-side configuration)

All Postgres-specific options go in the [settings] section:

```
[settings]
 sslmode=require
```

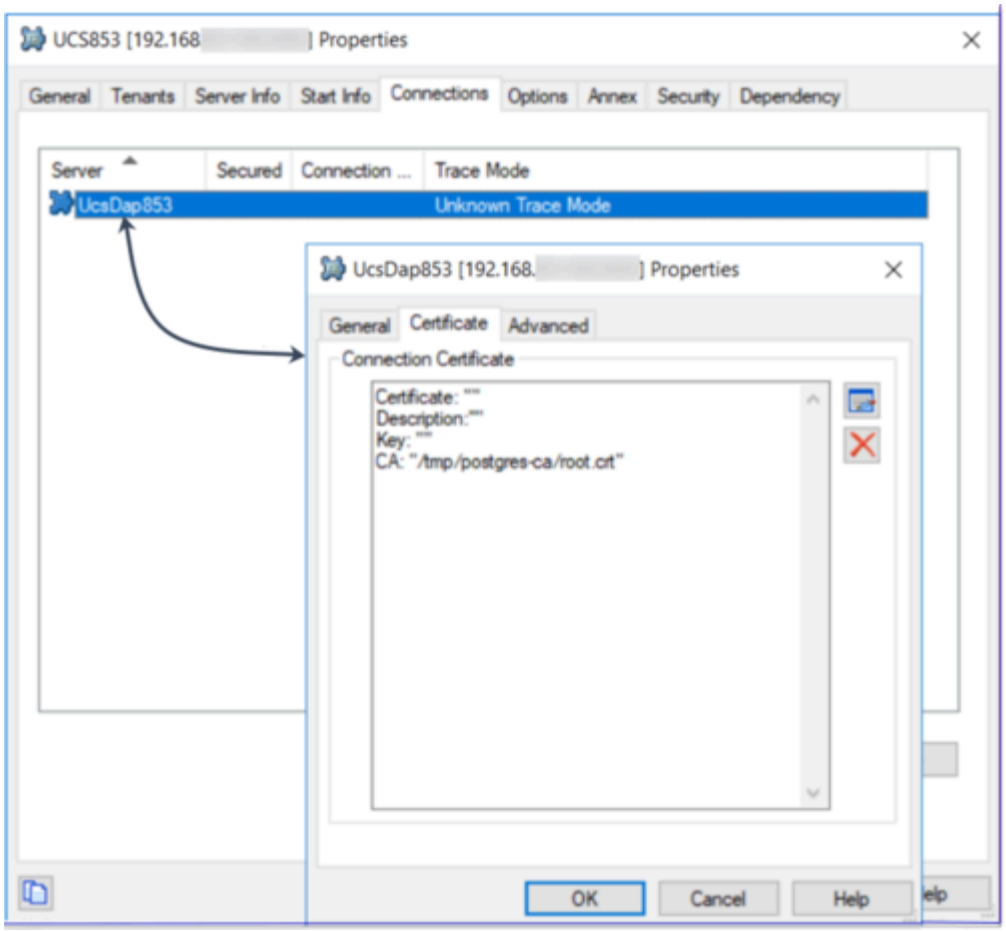
### Method 2

Alternatively, you can configure the application through the Certificate View, which synchronizes the **trusted-ca** option in the **[security]** section previously described. Note that other options still need to be manually added in the **[security]** section.




## Configuration on the connection link

You can configure the connection between the UCS application and the Postgres DAP application to inject TLS configuration and its FIPS mode.

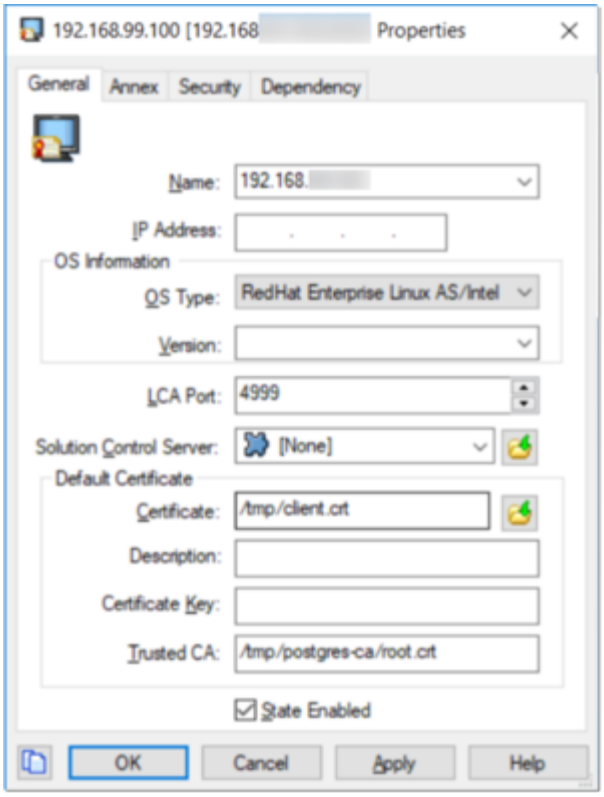



On the **Advanced** tab:

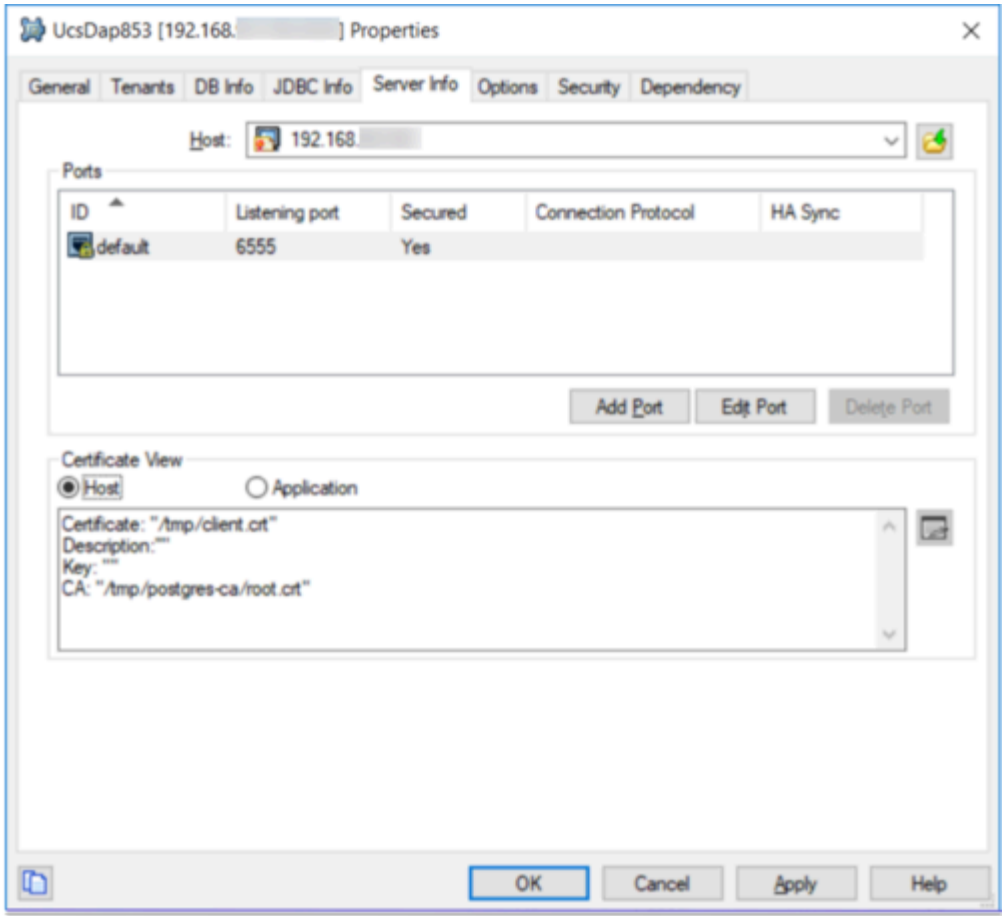
1. Set the **fips140-enabled** option to `true` to enable FIPS mode, which must be specified manually.
2. Specify any other other required parameters.

## Configuration on the Postgres host

The connection between the UCS application and the Postgres DAP application can be configured to inject TLS configuration and its FIPS mode.




The host information on the **Server Info** tab should look like this:




## Configuration hierarchy

UCS first looks in the UCS DAP application for TLS/FIPS configuration. If it is not found there, the connection between UCS and UCS DAP is scanned for TLS/FIPS information.

## Logs

Logs contain the TLS configuration retrieved (from which object) as well as the FIPS mode set:

```
16:54:59.894 Dbg 09900 [Ucs-Main ] <[]> Registering DataAccessPoint 'UcsDap853' with role
 'Main'
 16:54:59.900 Trc 09900 [Ucs-Main ] <[]> TLS flag enabled on application UcsDap853 (155)
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Expected Hostname : null
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Certificate : null
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Certificate Key : null
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Cipher List : null
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Certificate Revocation List : null
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Provider : null
```

```
16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Trusted CA Certificate : /tmp/postgres-ca/
root.crt
16:54:59.900 Std 21111 [Ucs-Main ] <[]> The Database Access Point 'UcsDap853' has been
configured with an infinite timeout, default 60 s will be used.
16:54:59.905 Int 09900 [Ucs-Main ] <[]> Ignoring invalid value 'null' for max-connections
```

## Further reading

Please consult the following sources:

- Genesys TLS configuration - Developer's Guide
- Postgres vendor documentation

# Setting up the UCS Database

**Purpose:** To set up the database or databases that UCS will use.

## Prerequisites

RDBMS, either Oracle or Microsoft SQL. See also the eServices 8.0 Deployment Guide. See the Deployment category page for overall prerequisites for deploying UCS.

## Procedure

1. Create a database in your RDBMS.
2. Locate scripts in `\Universal Contact Server\<application-name>\sql_scripts\<RDBMS-type>`.
3. Run `ucs-<RDBMS-type>.sql` for a new installation or choose the proper upgrade script for your RDBMS type.
   For an existing UCS database, run all scripts that cover your existing version, the current version, and all versions in between. For example, to upgrade from 7.6.1 to 8.0.2, you must run
   1. `upgrade_<RDBMS-type>_7.6.1_to_8.0.0.sql`
   2. `upgrade_<RDBMS-type>_8.0.0_to_8.0.1.sql`
   3. `upgrade_<RDBMS-type>_8.0.1_to_8.0.2.sql`

   Genesys supplies upgrade scripts for all releases starting with 7.0.1.

## Special Information for Oracle RAC

### DAP Configuration

To connect UCS to an Oracle Real Application Cluster (RAC), configure a DAP for UCS as follows:

- Use the first node's host and port settings on the `Server info` tab.
- For the host, port, and ONS settings of each additional node, create options in the `settings` section, as follows. Note that the ONS settings are optional.
  - Name: ONSConfiguration
    Value: nodes=node1:node1port,node2:node2port, ... where port is the ONS port, usually 6251
  - Name: hostx, where x is a positive integer
    Value: host of RAC
  - Name: portx, where x is a positive integer matching one of the hostx options

Value: DB port of RAC, usually 1521

- Create a service option to specify the network service name of the Oracle database.

Here is an example configuration for three nodes, named rac1, rac2, and rac3. The DB port is 1521 and the ONS port is 6251 for all nodes.

```
[ServerInfo]
host: rac1
ports: default, 1521
[Options > settings]
ONSConfiguration: nodes=rac1:6251,rac2:6251,rac3:6251
host1: rac2
port1: 1521
host2: rac3
port2: 1521
```

## UCP Library

Support of Oracle RAC also requires that you deploy the Universal Connection Pool library.

When connected to an Oracle RAC configuration, the UCS database layer uses the **ucp.jar** file for connection handling. If UCS is started against an Oracle RAC without the jar file, UCS will fail to start.

> ### Important
>
> Starting from version 8.5.300.36, UCS now provides the **ucp.jar** file as part of the installation package.
>
> The below section applies only for UCS versions prior to 8.5.300.36

To deploy the UCP library:

1. Download the `ucp.jar`file, version 12.2.0.1.0 (higher versions are not supported) from the Oracle web site.
2. Copy the jar file to the UCS home folder in `./lib/db/oracle`.

# Client-side Port Definition

eServices supports client-side port definition between components as outlined in the following tables. In these tables, the components listed in the first column are connecting to the components that are listed in the top row. Refer to the Genesys Security Deployment Guide for information about how to configure client-side port parameters.

## Client-side Port Support 1

## Client-side Port 1 Connections

| Connection | Config Server | Config Server Proxy | Solution Control Server | Message Server | Interaction Server | Universal Contact Server |
|---|---|---|---|---|---|---|
| **Interaction Server** | Yes | Yes | N/A | Yes | Yes | Yes |
| **Interaction Server Proxy** | Yes | Yes | N/A | Yes | Yes | N/A |
| **Universal Contact Server** | Yes | Yes | N/A | Yes | N/A | |
| **Universal Contact Server Proxy** | Yes | Yes | N/A | Yes | N/A | Yes |
| **E-mail Server** | Yes | Yes | N/A | Yes | Yes | Yes |
| **Chat Server** | Yes | Yes | N/A | Yes | Yes | Yes |
| **SMS Server** | Yes | Yes | Yes | Yes | Yes | N/A |
| **Social Messaging Server** | Yes | Yes | N/A | Yes | Yes | Yes |
| **Classification Server** | Yes | Yes | N/A | Yes | N/A | No |
| **Web API Java** | Yes | Yes | Yes | Yes | No | No |
| **Web API** | Yes | Yes | Yes | No | No | No |

| Connection | Config Server | Config Server Proxy | Solution Control Server | Message Server | Interaction Server | Universal Contact Server |
|---|---|---|---|---|---|---|
| **.NET** | | | | | | |

## Client-side Port Support 2

## Client-side Port 2 Connections

## TLS Support 2

| Connection | UCS Proxy | E-mail Server | Chat Server | SMS Server | Social Messing Server | Classification Server |
|---|---|---|---|---|---|---|
| **Interaction Server** | N/A | Yes | Yes | Yes | Yes | Yes |
| **Universal Contact Server** | N/A | N/A | N/A | N/A | N/A | N/A |
| **Universal Contact Server Proxy** | | N/A | N/A | N/A | N/A | N/A |
| **E-mail Server** | N/A | | N/A | N/A | N/A | N/A |
| **Chat Server** | N/A | N/A | | N/A | N/A | N/A |
| **SMS Server** | N/A | N/A | Yes | | N/A | N/A |
| **Social Messaging Server** | N/A | N/A | Yes | N/A | | N/A |
| **Classification Server** | N/A | N/A | N/A | N/A | N/A | |
| **Web API Java** | N/A | No | No | N/A | N/A | N/A |
| **Web API .NET** | N/A | No | No | N/A | N/A | N/A |

## Client-side Port Support 3

## Client-side Port 3 Connections

| Connection | Web API Java | Web API .NET | DB Server | Universal Routing Server | Stat Server |
|---|---|---|---|---|---|
| **Interaction Server** | N/A | N/A | No | N/A | Yes |
| **Universal Contact Server** | N/A | N/A | N/A | N/A | No |
| **Universal Contact Server Proxy** | N/A | N/A | N/A | N/A | N/A |
| **E-mail Server** | N/A | N/A | N/A | N/A | N/A |
| **Chat Server** | N/A | N/A | N/A | N/A | N/A |
| **SMS Server** | N/A | N/A | N/A | N/A | N/A |
| **Social Messaging Server** | N/A | N/A | N/A | N/A | N/A |
| **Classification Server** | N/A | N/A | N/A | N/A | N/A |
| **Web API Java** | | | N/A | N/A | No |
| **Web API .NET** | | | N/A | N/A | No |

> **Important**
> Training Server does not support the client-side port connection type.

# Deploying an E-Mail System in Secured Mode

This section describes how to configure an e-mail system to work in secured mode using TLS/SSL. This applies to POP3, IMAP4, and SMTP. The purpose is to generate and install a public/private key pair.



## Configuring TLS/SSL for E-mail Server

This section describes procedures for configuring your E-mail Server application to work with TLS/SSL.

### 1. Generating the .truststore file

**Prerequisites:** The corporate e-mail server is configured to work in secured mode.

**Steps:**

1. From the certificate on the Corporate E-mail Server, extract the public key. The following is an example of extracting a public key using keytool:

   ```
   keytool -export -v -alias hostname.example.com -file <certificate_name>.cer -keystore
   <certificate_name>.truststore
   -storepass <certificate_password> keytool -import -alias hostname.example.com -file
   <certificate_name>.cer -keystore client.truststore -storepass <certificate_password>
   ```

   At this point, the **client.truststore** file contains the public key.
2. Copy it to the host on which E-mail Server is running.

### 2. Modifying the E-mail Server startup command line on Windows

**Prerequisites:** The **.truststore** file has been created.

**Steps:**

1. Open **JavaEmailServerDriver.ini** in a text editor.
2. In the **[JavaArgs]** section, add the following: `-Djavax.net.ssl.trustStore=<path to certificate>`
3. Save and close the file.

### 3. Modifying the E-mail Server startup command line on UNIX

**Prerequisites:** The **.truststore** file has been created.

**Steps:**

1. Locate the E-mail Server startup file (**emailServer.sh**).
2. Open the file in a text editor and modify the startup command line so E-mail Server can locate the **.truststore** file. For example:

```
java -Djavax.net.ssl.trustStore=”<path to certificate>” --Xmx512M ....
```

3. Save and close the file.

### 4. Configuring E-mail Server's POP, IMAP, and SMTP Ports

**Prerequisites:** The **.truststore** file has been generated and E-mail Server's startup command line has been modified.

**Steps:**

1. In Configuration Manager or Genesys Administrator, open the properties for your E-mail Server application.
2. In the Options tab, locate the **[pop-client]** section for IMAP and configure the **type**, **port**, and **enable-ssl** options. For example:

```
[pop-client1]
type = IMAP
port = 993 (the default SSL port for IMAP)
pop-connection-security = ssl-tls
```

3. Locate the **[pop-client]** section for POP3 and configure the **type**, **port**, and **enable-ssl** options. For example:

```
[pop-client2]
type = POP3
port = 995 (the default SSL port for POP3)
pop-connection-security = ssl-tls
```

4. Locate the **[smtp-client]** section and configure the **port** and **enable-ssl** options. For example:

```
port = 465 (the default SSL port for SMTP)
smtp-connection-security = ssl-tls
```

5. Save your changes.
6. (Optional) If the application has already started, restart the application to apply the changes.

## Configuring TLS 1.2 for E-mail Server

**Prerequisites:**

- E-mail Server must run on JDK 1.8 or later.
- Ensure JDK 1.8 or later is installed. For example:
  - On the UNIX platform, the installed **emailServer.sh** (/usr/local/genesys/eservices/esj/emailServer.sh) must have JAVACMD pointing to JDK 8 or later, as follows:

    ```
    JAVACMD=/usr/lib/jvm/jre1.8.0_161/bin/java
    ```

  - On the Windows platform, the installed **JavaEmailServerDriver.ini** file must have the JVMPath reference to JDK 8 or later, as follows:

    ```
    JVMPath=C:\Programs\java\x64\jdk\jre\bin\server\jvm.dll
    ```

**Steps:**

1. Follow procedures in Configuring TLS/SSL for E-mail Server to generate a certificate with **trustStore**, to modify the E-mail Server startup command line on Windows and/or UNIX, and to configure E-mail Server's POP, IMAP, and SMTP ports.
2. Configure the following E-mail Server KVPs:

   ```
   [pop-client]
   mail.pop3s.ssl.protocols="TLSv1.2"
   (OR)
   mail.imaps.ssl.protocols="TLSv1.2"

   [smtp-client]
   mail.smtps.ssl.protocols="TLSv1.2"
   ```

   **Note:** The **pop-client** configuration must be added to all **pop-client-*** sections.

**Troubleshooting:**

To address possible issues on the Windows platform, consider completing the following steps:

1. Disable TLS 1.0 if necessary. See details in Microsoft documentation.
2. Enable TLS 1.2. See an example in this documentation.
3. Create an SSL self-signed certificate with **openssl**.
4. Add this self-signed certificate to the mail server (for example, set up an hMailServer with this certificate and add TCP/IP ports for IMAP-993, POP3-995, and SMTP-587 by following these instructions).
5. Verify the TLS version for the expected ports using **openssl**. For example:

   ```
   C:\> openssl s_client -connect <domainName>:993
   ```

   Example of the expected result:

   ```
   SSL-Session:
       Protocol  : TLSv1.2
       Cipher    : 0000
       Session-ID:
       Session-ID-ctx:
       Master-Key:
       Key-Arg   : None
       PSK identity: None
       PSK identity hint: None
       SRP username: None
       Start Time: 1526325099
   ```

```
Timeout   : 300 (sec)
Verify return code: 0 (ok)
```

## Configuring the Corporate E-mail Server

Configure TLS/SSL in the Corporate E-mail Server. Follow the constructor recommendations to generate a certificate and configure TLS/SSL on ports POP3, IMAP and SMTP.

The following is an example of generation of a certificate with keytool (keytool is a Java utility that is available with the JRE. The utility can be found in `<eServices_Install_Dir>/jre/bin` for UNIX operating systems, and in `<eServices_Install_Dir>\jre\bin` for Windows operating systems):

```
keytool -genkey -v -alias hostname.example.com -dname
“CN=hostname.example.com,OU=IT,O=ourcompany,C=FR” -keypass <certificate_password>
-keystore <certicate_name>.keystore -storepass <certificate_password> -keyalg “RSA” -sigalg
“SHA1withRSA” -keysize 2048 -validity 3650
```

The arguments used in this command are the following:

- `-alias`—Defines an alias in keystore, to store the key.
- `-dname`—Distinguished Name, a comma-separated list made up of the following, in the following order:
  - CN—Common Name. This must be the name of the host where the corporate e-mail server is running. It must be the host name used in E-mail Server's settings; for example, if connecting to a POP 3 server, the option server in the pop-client section must have this value.
  - OU—Organizational Unit Name
  - O—Organization Name
  - L—Locality Name (city)
  - S—State
  - C—Country Name

> **Important**
>
> - The abbreviations are not case-sensitive.
> - Only CN is required.

- `-keypass`—Password of the key of the certificate.
- `-keystore`—Specifies the keystore used.
- `-storepass`—Password of the keystore.
- `-keyalg`—Algorithm used to generate the key. Possible values are DSA and RSA. More information is available at `http://docs.oracle.com/javase`.

- `-sigalg`—Specifies the algorithm used to sign the key.
- `-keysize`—Specifies the size of the key.
- `-validity`—Defines the validity of the certificate, in days. The value in the example is 3,650 days, or 10 years.

# Multiple Interaction Servers in a Single Tenant

## Overview

If you deploy separate solutions within a single tenant, the usual architecture has a single Interaction Server processing all the interactions according to business processes that are defined for the entire tenant (which may include separate business processes for the separate solutions). The server also uses a single database to store all the interactions.

However, assigning a separate instance of Interaction Server to each solution can provide the following benefits.

### Functional Separation

- Stopping or terminating one solution does not affect the others.
- New solutions can be deployed with their own Interaction Servers without disturbing existing solutions.
- The Interaction Servers assigned to different solutions can be configured most appropriately to handle the specific solution.
- The overall system performance can be higher:
  - The servers do not affect each other and can utilize different hardware.
  - The servers use separate databases.
- Each solution can be managed and maintained separately by different teams.

### Data Separation

Physically separating the data of different solutions within one tenant provides the following benefits:

- Databases can be managed separately. Backup or restoration of one solution does not affect other solutions.
- Data corruption or hardware failures in one database only affects one solution.
- Performance is improved.

> **Important**
>
> If you have separate objects in a single tenant for use with multiple Interaction Servers, you must similarly use multiple instances of Stat Server, desktop

applications, reporting applications, and any other application that supports a connection to only one application of Interaction Server or T-Server type.

## Configuring Multiple Interaction Servers

You can use Configuration Server security to allow a specific server instance to work only with a subset of Business Processes within a tenant. At a high level, the procedure is as follows:

1. Use the **Interaction Design** window of Interaction Routing Designer to create an account and associate a Business Process with it. For more information, see the "Last Step in Creating a Business Process" topic of Universal Routing 8.1 Interaction Routing Designer Help (control of access to other configuration objects is not supported at this time).
2. Associate any other desired Business Processes with this account. This creates an `Access Group` object that has access to the associated Business Process.
3. In Genesys Administrator or Configuration Manager, create a Person object and add it to the Access Group.
4. Configure the selected Interaction Server instance to run under this new Person account:
    1. Configuration Manager: In the **Log On As** area of the **Security** tab of the Interaction Server Application object, select `This Account`, then select the desired Person in the resulting **Add User** dialog box.
    2. Genesys Administrator: On the **Configuration** tab of the Interaction Server Application object, go to the **Server Info** area, clear the **Log On As System** checkbox, and click **Browse** to select an account.

# Deploying an E-Mail Solution Using MIME Customization

Multipurpose Internet Mail Extensions (MIME) is a standard that allows e-mail messages to include graphics, audio or video files, or text in languages other than English. This section describes how to create a custom transformer that enables you to modify the content of e-mails and still ensure that they are compliant with MIME standards.

MIME customization in a Genesys e-mail handling environment ensures that you have normalized formats for all e-mail messages processed by your system and stored in the UCS database. To accomplish this, your e-mail solution uses a custom transformer to transform any MIME content in incoming or outgoing messages.

You implement this solution by using MIME customization APIs to change the content of incoming and outgoing e-mails. After creating the custom MIME transformer, you configure the E-mail Server Application object to enable the MIME customization option.

Your custom server then does all of the work required to handle incoming and outgoing e-mail messages as needed.

- When an e-mail message is received, E-mail Server uses the custom MIME transformer to transform the message before saving it in the UCS database.
- When an e-mail message is sent, E-mail Server uses the custom MIME transformer to transform the message (if it requires MIME customization) before sending it to the external e-mail server.

The following sections describe how to deploy an E-mail Solution using MIME.

## Deploy MIME Solution

## Deploying a MIME-compliant E-Mail Solution

## Viewing a sample transformer

E-mail Server includes a sample incoming MIME transformer. This sample is an "MS-TNEF Microsoft specific format to MIME" transformer, and can be found in the `mimeapi` subdirectory.

1. Open the directory where E-mail Server is installed.
2. In the `mimeapi` subdirectory, you can find the following files:

- `esj-mime-api-doc.jar`—Javadoc documentation for the API.
- `esj-mime-api.jar`—The actual API archive.
- `samples`—A subdirectory, containing the following files:
  - `TNEFMimeTransformer.java`—Java source code for this sample.
  - `readme.txt`—A readme file describing the sample.

> **Tip**
> Use the instructions in the `readme.txt` to download the JTNEF library and then compile the source. After you complete these two actions, you can use the sample to transform MS-TNEF incoming mails into regular MIME messages.

## Creating a custom transformer

To transform MIME content of e-mail messages, you must create a custom transformer. Your transformer code needs two Java classes: one to implement the API for transforming MIME content of incoming e-mail, and another to provide an API that transforms MIME content of outgoing e-mail. Both of these classes return the transformation result, and contain the following parameters:

- `input`—The MIME message content.
- `config`—Properties contained in the `[mime-custom-outbound-properties]` section of your E-mail Server Application object.
- `debugLogStream`—Log object to be used for debugging purposes.

### Each custom class that you create should satisfy the following conditions:

- Implements one of the two appropriate interfaces.
- Is thread-safe.

1. Create two Java classes: one for transforming incoming e-mail, another for transforming outgoing e-mail. Use the following interfaces:
   - Incoming E-Mail Customization API
   - Outgoing E-Mail Customization API
2. Bundle these two classes into a JAR file called `mimecustomization.jar`.
3. Place the new JAR file in the `esj\lib\external\` folder of your E-mail Server installation.

## Incoming E-Mail Customization API

Your custom transformer for incoming e-mail must implement the following interface:

### EmailInTransformer Interface

```
public interface EmailInTransformer {
  public TransformerResult transform(byte[] input, java.util.Properties config,
      java.io.PrintStream debugLogStream);
    public class TransformerResult {
      public static TransformerResult noTransformationNeeded();
      public static TransformerResult succesfull(byte[] transformedInput);
      public static TransformerResult failure(String failureReason, FailureAction
            failureAction);
     // Implementation details skipped
   }
   public class FailureAction {
      public static final FailureAction RETRY;
      public static final FailureAction BYPASS_TRANSFORMATION;
      public static final FailureAction DEPEND_ON_BAD_FORMAT_OPTION;
     // Implementation details skipped
   }
  }
}
```

## Outgoing E-Mail Customization API

The only difference between this API and the incoming e-mail customization API is that the `failureAction` parameter and class have been removed. If a transformation fails, then the original message is sent to the external e-mail server. Your custom transformer for outgoing e-mail must implement the following interface:

### EmailOutTransformer Interface

```
public interface EmailOutTransformer {
   public TransformerResult transform(byte[] input, java.util.Properties config,
       java.io.PrintStream debugLogStream);
   public static class TransformerResult {
       public static TransformerResult noTransformationNeeded();
       public static TransformerResult succesfull(byte[] transformedInput);
       public static TransformerResult failure(String failureReason);
  // Implementation details skipped …
       }
  }
}
```

## Configuring the E-Mail Server Application Object

After you create the custom MIME transformer, configure the E-mail Server Application object to specify the class names and configuration settings used with your custom transformer. E-mail Server will use the values you specify to transform content that is sent and received as e-mail messages. In

Configuration Manager or Genesys Administrator, configure the following sections on the Options panel of your E-mail Server Application object:

- `[mime-custom-inbound-properties]`—Content in this section is passed to the `EmailInTransformer.transform()` method using the config parameter. You can use this section to define custom options and settings.This section is not part of the default template. Add this section manually to set options for your custom transformer for incoming e-mail.
- `[mime-custom-outbound-properties]`—Content in this section is passed to the `EmailOutTransformer.transform()` method using the config parameter. You can use this section to define custom options and settings.This section is not part of the default template. Add this section manually to set options for your custom transformer for outgoing e-mail.
- `[mime-customization]`—Use this section to enable or disable the MIME customization, to specify the fully qualified class names of your custom classes, or to set allow optional debugging or saving features.

## MIME Options

## MIME Customization Options

The following table describes the options for the `[mime-customization]` Section of the E-mail Server application object.

| Option Name | Value | Description |
|---|---|---|
| enable-inbound | Default Value: `false`<br>Valid Values: `true`, `false` | If set to `true`, sends inbound e-mail messages to the specified inbound MIME transformer class. |
| enable-outbound | Default Value: `false`<br>Valid Values: `true`, `false` | If set to `true`, sends outgoing e-mail messages to the specified outbound MIME transformer class. |
| enable-inbound-debug-log | Default Value: `false`<br>Valid Values: `true`, `false` | If set to `true`, activates the inbound debug logger. |
| enable-outbound-debug-log | Default Value: `false`<br>Valid Values: `true`, `false` | If set to `true`, activates the outbound debug logger. |
| inbound-class-name | Default Value: ""<br>Valid Values: <any string> | Specifies the fully qualified name of the custom inbound transformer. |
| outbound-class-name | Default Value: ""<br>Valid Values: <any string> | Specifies the fully qualified name of the custom outbound transformer. |
| inbound-keep-received-mime | Default Value: `false` | Controls the way E-mail Server saves the content of MIME |

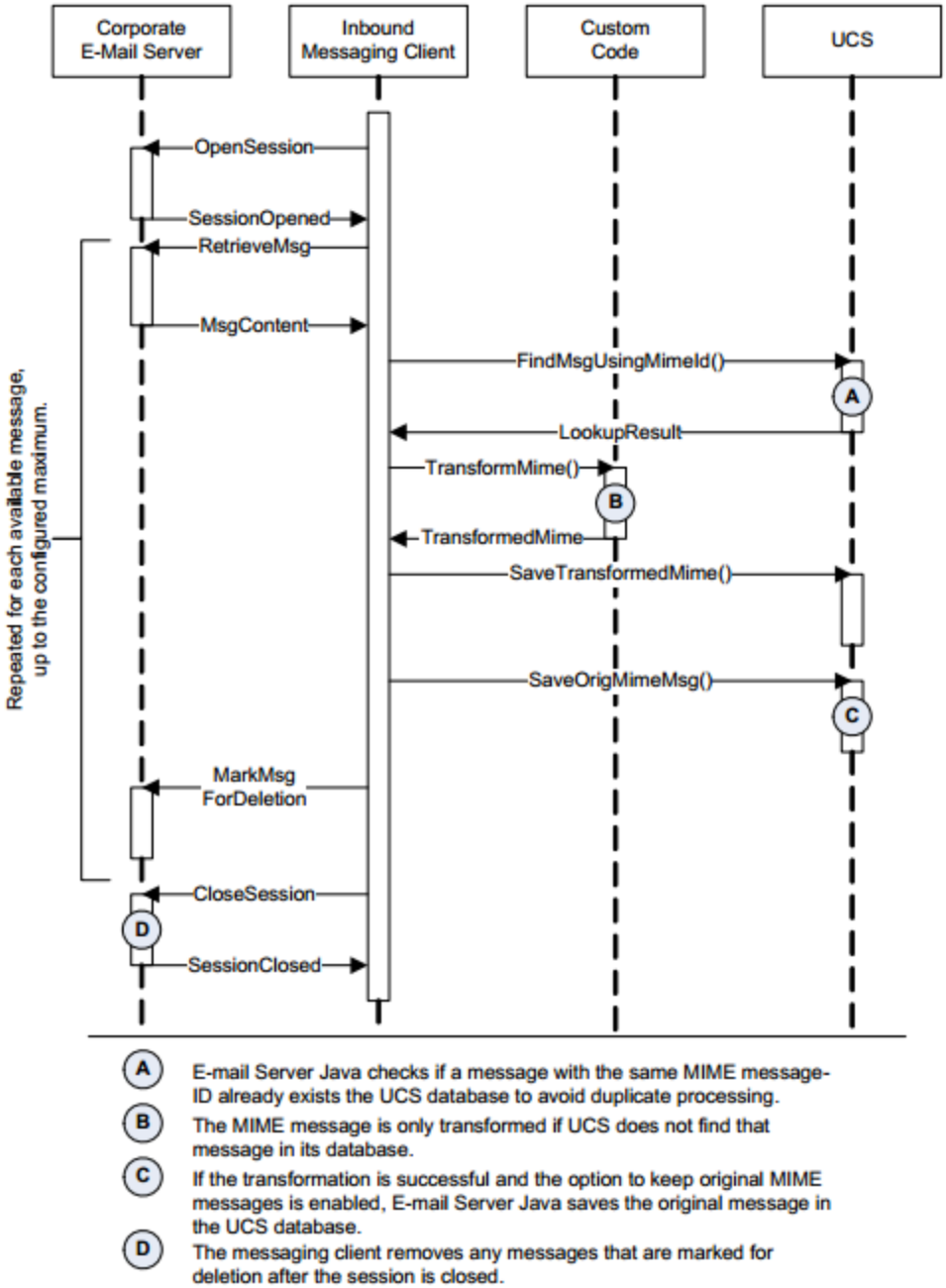
| Option Name | Value | Description |
|---|---|---|
| | Valid Values: `true`, `false` | messages.<br><br>If set to `true`, and if the message was transformed successfully, then saves the unmodified<br>MIME content of incoming e-mails received in the UCS database (along with the transformed content). |
| outbound-keep-sent-mime | Default Value: `false`<br><br>Valid Values: `true`, `false` | Controls the way E-mail Server saves the content of MIME messages.<br><br>If set to `true`, and if the message was transformed successfully, then saves the transformed<br>MIME content of outgoing e-mails is also saved in the UCS database (along with the initial content). |

## Message Flows

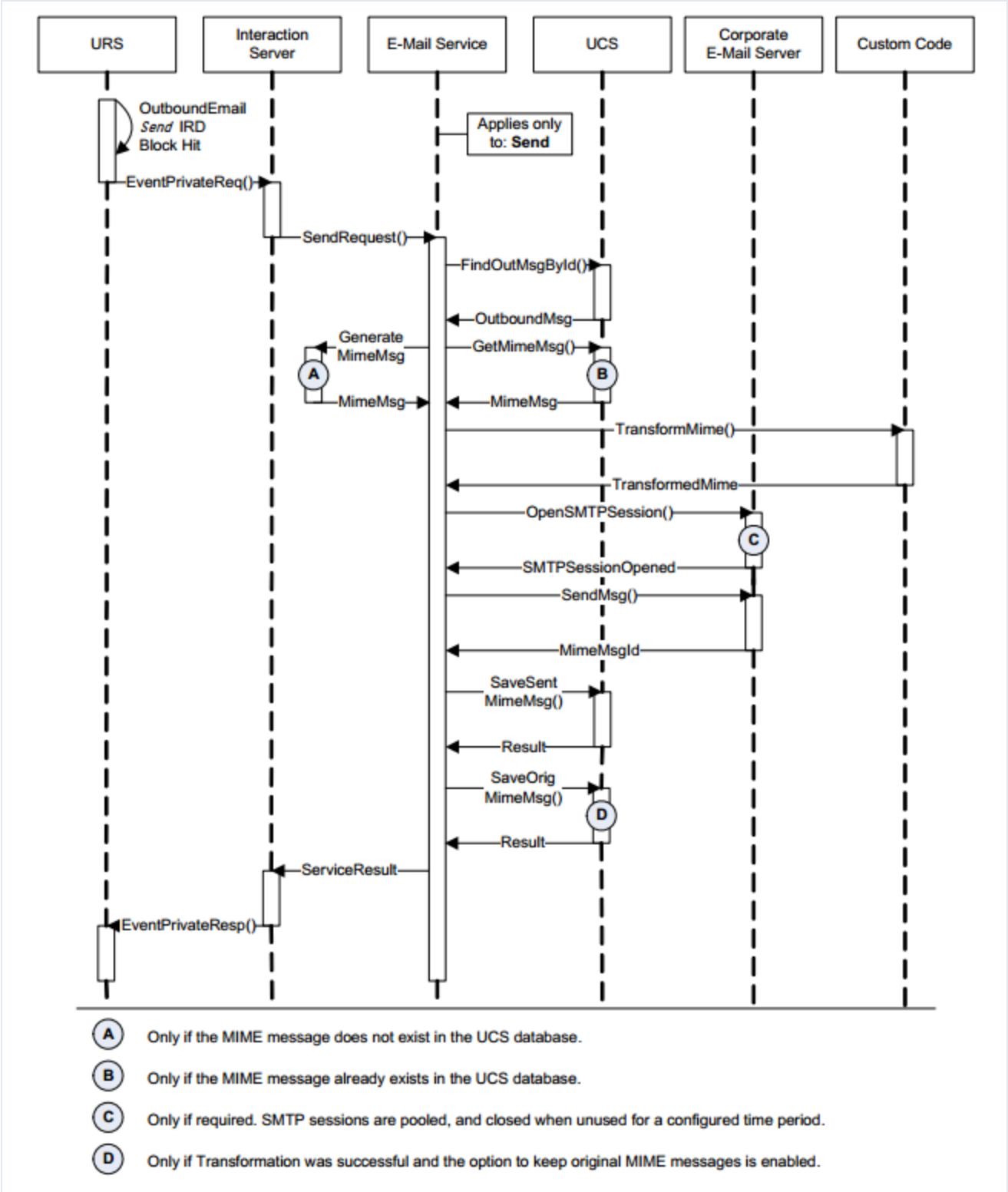
## Message Flow Patterns

The following diagrams show the message patterns for e-mail messages.

| Incoming E-Mail Message Flow |
|---|




**Outgoing E-Mail Message Flow**


URS
Interaction Server
E-Mail Service
UCS
Corporate E-Mail Server
Custom Code
OutboundEmail
Send IRD
Block Hit
Applies only to: Send
EventPrivateReq()
SendRequest()
FindOutMsgById()
OutboundMsg
Generate MimeMsg
GetMimeMsg()
A
B
MimeMsg
MimeMsg
TransformMime()
TransformedMime
OpenSMTPSession()
C
SMTPSessionOpened
SendMsg()
MimeMsgId
SaveSent MimeMsg()
Result
SaveOrig MimeMsg()
D
Result
ServiceResult
EventPrivateResp()
A Only if the MIME message does not exist in the UCS database.
B Only if the MIME message already exists in the UCS database.
C Only if required. SMTP sessions are pooled, and closed when unused for a configured time period.
D Only if Transformation was successful and the option to keep original MIME messages is enabled.

# Deploying Interaction Server 8.5.x

Apart from the information provided on this page, Interaction Server 8.5.x is deployed in the same way as in previous releases.

## UTF-8 Support

Interaction Server supports UTF-8 encoding on UNIX and, starting with release 8.5.1, on Windows platforms. Generally, the server follows the locale setting of the operating system: on UNIX platforms, the LANG environment variable; on Windows, the system locale in the Control Panel settings.

Starting with release 8.5.1, Interaction Server automatically detects the UTF-8 mode of Genesys Configuration Server and overrides its own locale to UTF-8 on all platforms. In this case, both Genesys Configuration Server and Interaction Server use UTF-8 encoding internally and for communication. For a non-UTF-8 Configuration Server, it is important to have the same system locale setting on all systems that run Genesys components in order for all components to use and communicate with the same ANSI or UTF-8 encoding.

It is possible, but not recommended, to override Interaction Server's locale setting by using the command line argument `-codepage <locale name>`. This might be useful in a situation, for example, in which all components use UTF-8 encoding except for a legacy, non-UTF-8 Configuration Server.

Note that Windows does not natively support the UTF-8 locale and it is not possible to set the system locale in the Control Panel to UTF-8. In this case, the command line argument `-codepage en_US.UTF-8` (or simply `-codepage UTF-8`) can be used to override Interaction Server's locale. Again, this is not necessary with a UTF-8 Configuration Server, since Interaction Server can detect that automatically.

## Database Connection

Interaction Server offers the following ways to connect to its database:

- With ODBC.
- (Interaction Server 8.5.1/8.5.2 only) With DB Server—This configuration is the one used in releases 8.5.0 and earlier. It makes use of two configuration objects, a DAP and a DB Server instance, and is described in earlier versions of this Deployment Guide and in the Management Framework documentation.

> **Important**
>
> Interaction Server 8.5.3 and higher does not support DB Server. Refer to Migrating to ODBC from DB Server for more information.

## Multiple Interaction Servers

There are two ways to do configure multiple Interact Servers within a single tenant:

- You can use Configuration Server security to allow a specific server instance to work with a subset of Business Processes within a tenant.
- Starting with release 8.5.106.x of Interaction Server and Interaction Server Proxy, you can also accomplish this using Interaction Server clusters.

# ODBC Connection

> **Important**
> The information on this page was current at the time of the Interaction Server 8.5.1 release. For Interaction Server 8.5.3, refer to the Migrating to ODBC from DB Server page.

These directions apply primarily to Linux. Directions on deploying ODBC drivers for Windows are available on this page in the Integrated Capture Points documentation.

## Install unixODBC

### Download

For compatibility purposes, the easiest and fastest way to do this is to download a prebuilt RPM or package that is compatible with your current Linux version.

> **Important**
> The Microsoft web page about the Microsoft ODBC Driver 11 for SQL Server on Linux refers to only version 2.3.0 of unixODBC Driver Manager.

You can also

- Download, build and install unixODBX from the source provided at http://www.unixodbc.org/.Follow the instructions on the page.
- Build your own rpm. Information is at https://fedoraproject.org/wiki/How_to_create_an_RPM_package

> **Warning**
> Genesys has no responsibility or license for unixODBC, which is a third-party package. Any and all of these unixODBC installation options are merely recommendations, and it is your responsibility to the choose one that fits your purpose best. There are dependencies for each of these options. Also, environments with alternative third-party drivers, database accelerators and ODBC Managers are not supported on compatibility issues. Genesys recommends that you not use ODBC drivers for different RDBMS (i.e. ORACLE and DB2) on UNIX platforms simultaneously.

### Check

Once you have downloaded and installed unixODBC, check the **odbcinst.ini** (under **/etc** in the installation directory) and **odbc.ini** (in **$HOME/.odbc.ini**) files. This guide will refer to the unixODBC installation directory as the environment variable $UNIXODBC.

## Install ODBC drivers

### Oracle

#### Install the ODBC Driver

1. Download the latest version of the ODBC driver from http://www.oracle.com/technetwork/topics/linuxx86-64soft-092277.html. You will need these two packages:
   - Instant Client Package - Basic: All files required to run OCI, OCCI, and JDBC-OCI applications
   - Instant Client Package - ODBC: Additional libraries for enabling ODBC applications
2. Unzip both packages to the same directory, which this guide will refer to as environment variable $DRIVER.
3. From the $DRIVER directory, run the following:

   ```
   ./odbc_update_ini.sh $UNIXODBC  $DRIVER [<Driver_Name>] [<DSN>]
   ```

   For example,

   ```
   ./odbc_update_ini.sh $UNIXODBC $DRIVER oracleodbc-12.1 ora
   ```

4. Set the three environment variables ORACLE_HOME, LD_LIBRARY_PATH, and TNS_ADMIN.
   - TNS_ADMIN is the directory containing the **TNSNAMES.ora** file, which describes Oracle connections. If such a file does not exist, you must create it.
   - ORACLE_HOME is the directory where the **bin** and **lib** directories are located for the Oracle client.
   - LD_LIBRARY_PATH is the load library path.

   For example,

   ```
   export ORACLE_HOME=$DRIVER
   export LD_LIBRARY_PATH= $UNIXODBC/lib/:$DRIVER/
   export TNS_ADMIN=/etc/oracle
   ```

#### Sample

This is a sample of **odbc.ini** with a configured DSN:

```
[ora]
Application Attributes = T
Attributes = W
BatchAutocommitMode = IfAllSuccessful
BindAsFLOAT = F
CloseCursor = F
```

```
DisableDPM = F
DisableMTS = T
Driver = OracleODBC-12.1
DSN = ora
EXECSchemaOpt =
EXECSyntax = T
Failover = T
FailoverDelay = 10
FailoverRetryCount = 10
FetchBufferSize = 64000
ForceWCHAR = F
Lobs = F
Longs = T
MaxLargeData = 0
MetadataIdDefault = F
QueryTimeout = T
ResultSets = T
ServerName = GENESYS_INX
SQLGetData extensions = F
Translation DLL =
Translation Option = 0
DisableRULEHint = T
UserID = SYSTEM
Password = system
StatementCache=F
CacheBufferSize=20
UseOCIDescribeAny=F
MaxTokenSize=8192
```

> **Important**
>
> If Interaction Server uses ODBC to connect to the Oracle database in your environment, you must set the option **Lobs** to T in the **odbc.ini** file.

This is a sample of **odbcinst.ini:**

```
[OracleODBC-12.1]
Description=Oracle ODBC driver for Oracle 12g
Driver=/usr/lib/oracle/12.1/client64/lib/libsqora.so.12.1
Setup=
FileUsage=
CPTimeout=
CPReuse=
Driver Logging=7
```

In this example, the DSN name is `ora`. ServerName is the corresponding service name in **TNSNAMES.ora:**

```
GENESYS_INX =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = fakehost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = genesys_inx)
    )
  )
```

## MS SQL

> **Important**
> MS SQL officially supports only version 2.3.0 of unixODBC

1. Run

   ```
   export LD_LIBRARY_PATH= $UNIXODBC/lib
   ```

2. Follow the installation instructions given at http://www.microsoft.com/en-us/download/details.aspx?id=36437
   If you want to use the MSSQL Driver with unixODBC version 2.3.1 or greater, you must perform the installation with the flag `-force`, and afterwards you must do the following:

   1. Locate the file **libodbcinst.so.2** that was installed by unixODBC.
   2. Make a symbolic link to it:

      ```
      sudo ln -s libodbcinst.so.2 usr/lib64/libodbcinst.so.1
      ```

   **Odbcinst.ini** should now contain the following:

   ```
   [ODBC Driver 11 for SQL Server]
   Description=Microsoft ODBC Driver 11 for SQL Server
   Driver=/opt/microsoft/msodbcsql/lib64/libmsodbcsql-11.0.so.2270.0
   Threading=1
   UsageCount=1
   ```

3. Append a DSN entry name to **odbc.ini,** for example:

   ```
   [sqltest]
   Driver = ODBC Driver 11 for SQL Server
   DSN = sqltest
   Trace = No
   ServerName = fakehost
   UserID = genesys
   Password = genesys
   Database = genesys_inx
   ```

## DB2

### Install the ODBC Driver

1. Download the IBM Data Server Driver for ODBC and the CLI (64-bit) driver for 64-bit operating systems from http://www-01.ibm.com/support/docview.wss?uid=swg24029746
2. Decompress the driver package in the desired location (for example: **/usr/local/db2odbc**). We will refer to this location as `$DRIVER`.
3. Add the driver manually by editing the **odbcinst.ini** file to add the following:

   ```
   [DB2]
   Description=DB2 ODBC Driver
   Driver=$DRIVER /odbc_cli/clidriver/lib/libdb2o.so
   ```

4. Save the file.

### Test the Connection

1. Create a new DSN:
   1. Create a file **db2cli.ini** in **$DRIVER/odbc_cli/clidriver/cfg**
   2. Add this section to it:

      ```
      [SAMPLE_ODBC_DSN_NAME]
      hostname=your_db2_server_host
      port=your_db2_server_port
      database=name_of_the_DB
      protocol=TCPIP
      autocommit=0
      ```

   3. Save the file
2. Now you must add an entry to **odbc.ini** so unixODBC will know about that particular DSN: to do this, add the following section to `odbc.ini`:

   ```
   [SAMPLE_ODBC_DSN_NAME]
   Driver=DRIVER_NAME_FROM_ODBCINST.INI
   ```

   In our case the driver name is DB2 (see above), and the line will look like

   ```
   Driver=DB2
   ```

Detailed installation instructions are at https://www-304.ibm.com/support/docview.wss?uid=swg21418043

### PostgreSQL

Information about PostgreSQL is provided in the Framework Database Connectivity Reference Guide and at http://www.postgresql.org/.

## Configure the DAP

This section applies to Interaction Server and Event Logger. Related information about Database Capture Point is in a separate location.

For an ODBC connection, you must configure the Database Access Point (DAP) associated with the Application in question, as follows:

1. Add an option called **dbprotocol** to the **[settings]** section of the Interaction Server DAP and give it the setting `odbc`. For the Event Logger DAP, the equivalent is the **delivery-protocol** option in the **[logger-settings]** section. This forces Interaction server to use ODBC to access the databases.
2. In Genesys Administrator, enter the following in the **[DB Info]** section of the **Configuration** tab (in Configuration Manager, on the **DB Info Tab**):
   - DBMS Name
   - DBMS Type

- Database Name
- User Name
- Password

3. Optionally, specify a non-default driver, as described below.

To make switching to ODBC easier for existing installations, Interaction Server will try to build the ODBC connection strings itself based on the information that is available in the DAP. During this process, Interaction Server uses the following default names for the drivers:

| Database Type | Default Driver |
|---|---|
| MS SQL | {SQL Server Native Client 10.0} |
| DB2 | {IBM DB2 ODBC DRIVER - DB2COPY1} |
| Oracle | {Oracle in OraClient11g_home1} |
| PostgreSQL | {PostgreSQL ANSI(x64)} |

If your configuration uses a different driver name (as in the examples of Oracle and DB2 above), you must provide the actual driver name:

1. Locate or create the **connection-string** option in the **[settings]** section.
2. The value of **connection-string** is a list of key-value pairs separated by semicolons ( ; ).
   - To specify the actual driver, give it a value of `driver=<driver_name>`.
   - If your database server is not running on the default port (1433 for MSSQL, 1521 for Oracle and 50000 for DB2), you must also provide the port number by adding `port=<actual_port>` to the the value of **connection-string**; for example, `driver=DB2 ODBC Driver;port=1234`.

### DSN

You can also have Interaction Server use a DSN that has been configured in your system. To do this, locate or create the **connection-string** option in the **[settings]** section of the DAP (the **[logger-settings]** section in the Event Logger DAP), and set it to `DSN=<name_of_the_dsn>`. With this method, the user name and password to connect to the database are taken from the settings on the **DB Info** tab and do not need to be provided in the DSN properties.

## Test the Connection

To test the connection, run the command

```
Isql -v [DSN name]
```

If the proper credentials are not configured in the **odbc.ini** file, you may have to add them as parameters of this command; for example,

```
Isql -v [DSN name] [username] [password]
```

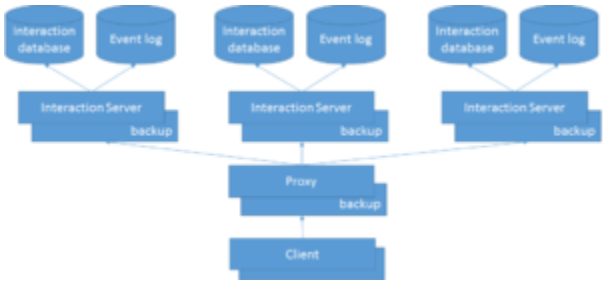
Related information can be found on the pages linked to this page in the Integrated Capture Points

documentation.

# Interaction Server Cluster

Starting with release 8.5.106.x of Interaction Server and 8.5.107.x of Interaction Server Proxy, you can configure multiple Interaction Servers into a cluster that works with a single instance of Interaction Server Proxy.

## Cluster Deployment Overview




Interaction Server Cluster Deployment

Each server node in an Interaction Server cluster has its own interaction database and its own event log databases (or other types of Event Logger). And each node must be configured in pairs, with one server configured as primary and the other as backup.

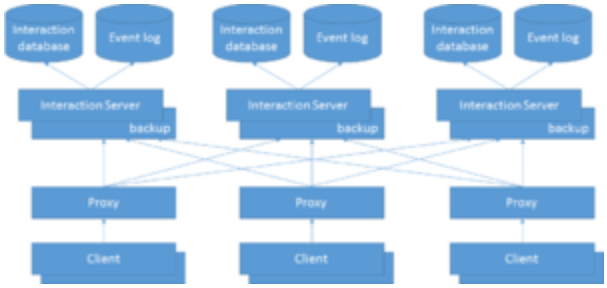
Interaction Server Proxy connects to all of the Interaction Server pairs in the cluster, presenting them to its clients as a single Interaction Server application. This includes presenting a single, consistent reporting event stream to reporting clients.

The proxy routes requests to the appropriate server or servers within a cluster, based on the cluster configuration and on the interaction identifier or agent specified in the request.

The proxy also load-balances requests, where appropriate, based on the list of tenants supported by each Interaction Server, the list of media types supported by each server, and other request attributes.

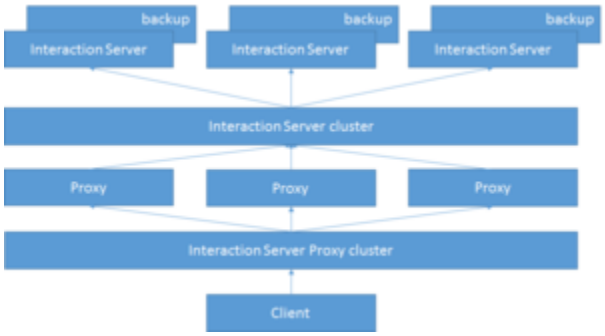
By dividing the incoming stream of interactions between multiple servers and databases, the cluster can be scaled to support a greater number of interactions than a single Interaction Server with a single database.

## Cluster Deployment with Multiple Proxies




Cluster Deployment with Multiple Proxies

Interaction Server Proxy does not maintain a persistent state, meaning that it can assume a specific state by making the appropriate requests to the appropriate Interaction Server nodes in the cluster. Because of this, you can set up several proxies working with the same cluster of Interaction Servers and each serving different clients. By using multiple proxies and distributing the clients between them, you can support a greater number of clients than you can with a single proxy.



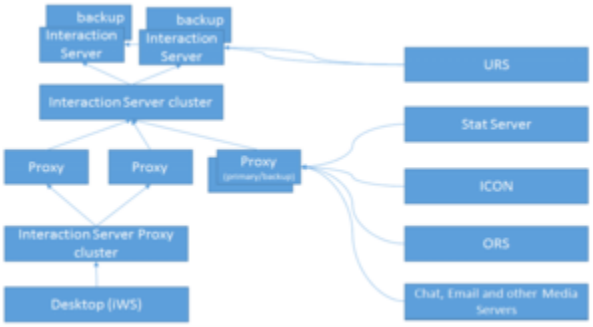

Interaction Server Cluster Configuration

Use the following two additional configuration objects of Application Cluster type to simplify cluster configuration:

- **Interaction Server cluster**:
  - The Interaction Server cluster configuration object holds the connections to all of the Interaction Server configuration objects in the cluster.
  - All of the proxy applications can share the same cluster configuration.
  - All proxies use the Interaction Server cluster configuration to connect to **all** Interaction Servers in the cluster **simultaneously**.
  - The proxy implementation can use either:
    - The Interaction Server cluster
    - Direct connections to Interaction Servers
    - Or a combination of the two
- **Interaction Server Proxy cluster**:

- The Interaction Server Proxy cluster configuration object holds the connections to all of the Interaction Server proxy configuration objects.
- All client applications can share the same proxy cluster configuration.
- The client applications use the Interaction Server Proxy cluster to choose one proxy randomly and are connected to a single proxy at any given moment.

## Suggested Deployment Configuration




Suggested Cluster Deployment Configuration

### Special Considerations

#### Media Servers

Media servers (including chat and e-mail) should be connected to a primary/backup pair of Interaction Server Proxies. Depending on the expected load, media server can share a pair of Proxies with other applications or can use a dedicated pair.

#### Genesys Desktop Apps

When using an Interaction Server Cluster with Genesys desktop applications such as Workspace Desktop Edition, you may use an N+1 architecture, where each Interaction Server Proxy configured for the Cluster of Proxies should only be run as a single application, with no backup instance. A desktop application configured with a connection to the Proxy Cluster will choose any available Proxy in the cluster.

#### Reporting and Orchestration

Separate primary/backup proxy pairs should be configured specifically to be used by reporting clients (Stat Servers and ICON) and, if appropriate, Orchestration Server (ORS). For reporting clients and ORS, you can deploy additional primary/backup proxy pairs, as needed.

#### Universal Routing Server

Universal Routing Server (URS) should be connected directly to all primary Interaction Servers in the cluster.

### Configuration Recommendations

Genesys recommends the following settings for Interaction Server nodes:

- login-session-timeout (this option was introduced in release 8.5.107)—1440, the default.
- not-ready-on-invitation-timeout—false.
- agent-login-control—Set to the same value on all Interaction Servers in all nodes.
- allow-multiple-agent-connections—false, in order to avoid the significant increase in the flow of notification events that Agent Desktop applications operating in a Cluster environment may experience if they are allowed to create multiple connections to Interaction Server for the same Agent/Place.

## Step by Step Cluster Configuration Process

1. Create the primary/backup pair of Interaction Servers with the required Database Access Point, following the process for a single server deployment.
2. Repeat Step 1 as many times as needed, keeping in mind that a single node within the cluster will be processing some dedicated solution or media type (for example, iWD or e-mail channel) or will be processing some media type in collaboration with other nodes.
3. Create Interaction Server Proxy applications using the Interaction Server application type, following the process for single Interaction Server Proxy deployment.
4. Repeat Step 3 as many times as necessary. Keep in mind that you can always add additional Proxy applications later if necessary.
5. At this point you can use Cluster Manager Plug-in for GAX to create clusters and assign media types, then continue with Step 6 below. If you are not using Cluster Manager, carry out Steps (a)–(f), then continue on to Step 6.
    a. Optionally, create a cluster-settings section in the Interaction Server application objects and add the media-types option with the value consisting of a comma-separated list of media types supported by the server. If this option is not present, it is assumed that Interaction Server can process any media type defined for the tenant. These options are not used by Interaction Servers, but by Interaction Server Proxies to correctly route interaction related requests to the servers in the cluster. Interaction Server Proxy does not use the list of tenants configured for it, but the list matters for its clients, which connect to the Proxy whose configured tenants match their own.
    b. Create the Interaction Server cluster application using the Application cluster application type. In its **[cluster-settings]** section, set the **cluster-type** option to Interaction Server.
    c. Add connections to all primary Interaction Servers you have created in Steps 1 and 2 to the connections of the new cluster application.
    d. Create the Interaction Server Proxy cluster using the Application cluster application type. In its **[cluster-settings]** section, set the **cluster-type** option to Interaction Server Proxy.
    e. Add connections to all primary Interaction Server Proxy applications you have created in Steps 3 and 4 to the Interaction Server Proxy cluster that you created in Step (d).
    f. For each Interaction Server Proxy application added to the Interaction Server Proxy cluster in step (e), add a connection to the Interaction Server cluster created in step (b)
6. Add connections to the Interaction Server Proxy cluster to all applications that support Interaction Server Proxy cluster configuration (for example, Desktop). Genesys recommends that you not use

primary/backup deployment for Proxy applications that will be serving client applications that support Proxy clusters.

7. Add connections to the dedicated Proxy for applications that do not support Interaction Server Proxy cluster configuration. Genesys recommends that you use primary/backup deployment for Interaction Server Proxy applications that serve such clients.
8. If you are using URS (rather than ORS), add connections to all primary Interaction Servers to the URS application. URS should be directly connected to each Interaction Server in the cluster.
9. If you are using ORS, add a connection to Interaction Server Proxy to ORS application.
10. For the Stat Server application there are two options: it can be connected to Interaction Server Proxy or directly to each Interaction Server in the cluster.

# Other Information

## Snapshots

Starting with release 8.5.107.x, Interaction Server clusters support snapshots.

The number of interactions in a snapshot can be limited either by the number specified in the request or by the Interaction Server option max-interactions-per-snapshot. Note that these limits are per server, not per client snapshot, so the number of interactions returned in a client snapshot may be more than any specified limit.

## Timeout for Pull for Strategy

You can tune the way that the cluster responds to pull requests from routing clients.

- For each Interaction Server in the cluster, set a timeout using the **[settings]/pull-hold-timeout** option.
- Use the Proxy option of the same name to set a timeout that overrides the setting of any Interaction Server instance in the cluster.

Setting the Proxy option to a relatively low value can help speed up operation in situations in which one server has no interactions and the Proxy must wait for its response before going on to the next server in the cluster.

# Cluster Limitations

## Nodes Are Irreplaceable

Interactions submitted to a given Interaction Server node cannot be processed by any other node.

Therefore, if you want redundancy of any kind, you have to deploy each node as a primary/backup pair, which gives you the level of reliability provided by the standard Genesys primary/backup functionality.

**Important:** A node for which both primary and backup Interaction Servers are down, or whose single node database is down, cannot process interactions, and no other nodes can complete any interactions that were in progress on this node. This also applies to the node's event log. If the event log database is not accessible for a significant time period, then events are lost.

## Interaction Queue FIFO Issue

Because each primary/backup pair works with its own database and the proxy requests interactions from the same queue in a round robin fashion, the order of interactions could be broken. This limitation is more important for long-lived interactions and large backlogs. For deployments with low volume and no backlog it is not very important.

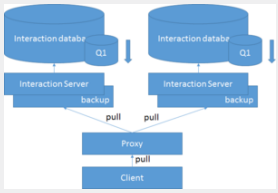


Diagram 1

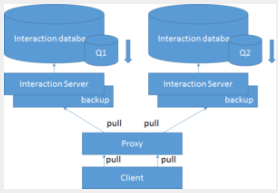


Diagram 2

In Diagram 1, two nodes handle the same type of interactions using the same business process and both handle the same queue Q1. Both servers maintain the ordering of the interactions, and respond to pull requests by the proxy in the correct order. But the order as seen by the client may not be correct, as the proxy has no way of tracking or handling the order correctly (it does take account of the semantics of order definition) and, therefore, cannot request the appropriate server (for example, to get older interactions or interactions with higher priority as defined by the interaction View order).

The same is true for URS, where each node pushes interactions from its own database in the correct order, but the order of routing of interactions by the URS is different. This also applies to the segmentation feature of an interaction View.

In certain situations, for example, when URS and interaction View use the same order by priority, URS can rectify the situation by processing interactions internally in the correct order, and, therefore, allowing one of the servers to push more interactions with higher priority.

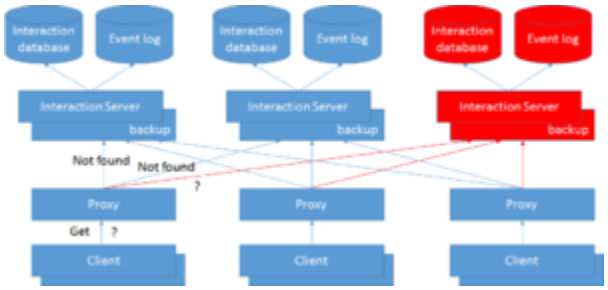
If URS or ORS or any other client that pulls interactions from an interaction View does not impose its own order of processing, but relies solely on the order provided by Interaction Server (or, in this case, by the proxy), the imbalance between the servers may increase in time, because there is no synchronization point unless the queues become empty. In other words, the situation with broken queue order may be acceptable for short-lived interactions, like chat, and less acceptable for long living interactions, like email.

Obviously, if different nodes process different types of interactions, and as a result, only one server in the cluster maintains the order of some specific queue, there is no such problem. For example, in Diagram 2, the client pulls from different queues defined for different solutions and both streams are provided in the correct order.

## Duplicate Interaction IDs

Because clients can provide their own interaction IDs, it is possible that interactions with the same ID may be submitted to different nodes at some point when the cluster cannot check for duplicates. This issue does not arise if interaction identifiers are generated by the cluster itself (by any node in the cluster) or by eServices components (since they all follow a schema that provides uniqueness within a single configuration environment). But if customers wish to generate IDs themselves, they must guarantee that the IDs are unique.

## No Definite Reply to Some Operations Regarding Interaction Existence



Interaction is not accessible

If some node in the cluster is down (even for a short time, for example, due to switchover or host restart) the proxy cannot definitely tell if some interaction exists or not. In this case, the proxy generates a new error (`Interaction is not accessible`) when it cannot contact all the nodes in the cluster. In the figure at right, the first and second nodes provide negative replies, but the proxy receives no reply from the third node, and therefore the proxy cannot definitely tell the client that the interaction is not found.

In certain cases, a proxy might already have information about which server processed an interaction with a specific ID in its cache. In this case, the proxy only contacts the server that it has identified as having processed the interaction, and even if some nodes are down, it still may reply with the definitive answer (`Interaction does not exist`). But, generally, the proxy may restart or a new proxy can be instantiated, and in that case its cache is initially empty and it still needs to broadcast to all servers to try and find the interaction.

## Limited Scope for URS

Because URS is connected directly to each primary/backup Interaction Server in the cluster, it only sends requests (including ESP requests) to the server it received the `EventRouteRequest` from, and this limits the scope of these requests. For example, the ESP request `FindInteraction` can only access interactions in the context of a single node. However, in many situations this might be acceptable since the context of such URS requests is logically limited to the same scope.

## Increased Limits for Some Operations

For certain requests, for example, `RequestFindInteractions`, `RequestGetWorkbinContent`, that use configured limits for the maximum number of interactions that Interaction Server can return in request, those limits are not enforced in a cluster configuration. This is because, by design, the proxy forwards the same request to all relevant servers and then merges the replies. In the worst-case scenario, the number of returned interactions might be a multiple of the number of nodes.

In some specific scenarios, such as searching by external id, a short set of data from only one node may be sent back to the Desktop client.

However, if only one node handles a specific media type, then limits are enforced just as they are with a single Interaction Server deployment.

## Incorrect Order of Interactions in Some Requests

The `EventWorkbinContent` or `EventInteractionsFound` event, in cases when multiple servers return the results, contain interactions out of order after the proxy merges the results. This is because ordering and filtering are handled by the database and not by the server or proxy, which do not take account of the semantics of the condition and order. The desktop should sort the result in this case.

## Event Log Limitations

There are two options for configuring the event logs in the cluster:

1. Configure one Event Logger database for the entire cluster and let all servers in the cluster write to this single database.
2. Configure separate Event Logger databases for each primary/backup pair in the cluster and then process multiple databases at some point later.

The first option has some limitations:

- Since each server writes its own pair of events `EventAgentLogin`/`EventAgentLogout`, if some triggers are implemented in the database (including standard triggers that remove agent events on `EventAgentLogout` either with or without delay) then the first `EventAgentLogout` event removes all the event records while agents may still be logged in on other servers. This situation is rarely a problem however, since the proxy logs agents in on all servers and logs them out on all servers at the same time.
- If an ETL or any consumer of event logs is to handle agent sessions correctly, it must take into account the multiple overlapping pairs of EventAgentLogin/EventAgentLogout.

Note that since specific interactions are processed by a specific server, the issue of overlapping event pairs does not exist for interactions or ESP reporting events.

The second option requires event log consumers to process multiple databases and merge the event streams taking into account the same problem of overlapping agent state events.

## Licensing

- Each server in the cluster checks out the number of licenses specified in its configuration. Each primary/backup pair in the cluster uses its own unique license group ID (DBID of the primary server) and only shares the licenses between the pair.
- Distribution of licenses:
  - Different nodes do not share licenses and the entire cluster requires the number of licenses equal to the sum of all configured licenses for every primary/backup pair.
  - The lowest number of licenses that is specified in configurations of nodes defines the capacity of the

cluster. An unsuccessful login to one node (due to exceeded number of licenses) revokes that agent's logins from other nodes in the cluster.

As an example, suppose you have 100 available agent-seat licenses and four nodes. If you set **ics_multi_media_agent_seat** to 30 in every node, then three nodes will allocate 30+30+30 licenses, and the fourth node will allocate the remaining 10. Then with all nodes running, 10 will be the maximum number of agents who can log in to the Cluster.

- A dynamic decrease of licenses on one node may result in logout of agents from the cluster (if the new limit exceeds the number of agents currently logged on to this node).

## Capture Points

Capture Points are configured per node and the limitation is that each Capture Point can only access interactions that are processed by the node for which it is configured. That is certainly acceptable for a configuration in which nodes are configured to handle a specific media type/solution exclusively, and no two nodes handle the same media type. In this case, all Capture Point operations should work, including operations that request information from the server.

In configurations in which there are multiple nodes that handle the same media type and each node has a Capture Point configured, these Capture Points can be used to capture interactions and produce notifications towards external systems, provided they can work with the source in parallel. This might be true for JMS Capture Points that are configured to connect to the same JMS queues, for File Capture Points that are configured to monitor different directories, and for Web Service Capture Points for which some sort of load balancing is configured.

Because each Capture Point is limited to visibility of only one node, requests that are intended to request data from Interaction Server might not work correctly. For example, the request to Get Interaction/Task properties might be received by a Capture Point other than the one that created the interaction: the request would fail, even though the interaction might still exist in the database of the other node.

## Stat Server Java Extensions

In a cluster deployment, SSJE can only receive data from one Interaction Server.

## Proxy Disconnect/Mass Agent Logout/Panic Signal

The functionality implemented in Interaction Server to shield reporting clients from massive agent logout due to proxy disconnect does not work when a proxy application is connected to Interaction Server Proxy. The server is not notified of the disconnection of a proxy that is connected to Interaction Server Proxy. The server is only notified of individual disconnect events for each client that was connected to or handled by the disconnected proxy. Because of that, the server has no information on how many clients are disconnected and cannot tell when the threshold is reached that would generate the corresponding panic signal to reporting clients.

This can be mitigated by connecting Stat Server to all Interaction Server in the cluster directly. Stat Server has the same functionality of aggregating the reporting event streams from multiple servers as Interaction Server proxy does.

## ESP Configuration Must Be Symmetrical

All servers serving the same tenant and media type should have connections to the same ESP servers to avoid a situation in which one Interaction Server can execute a particular ESP request and another cannot.

## System Monitoring/Ping Request

The proxy does not support Ping requests. The components that monitor Interaction Server have to be connected directly to the individual Interaction Servers in the cluster.

## Disabled Media Type

Unlike Interaction Server, which permits submission of interactions by media server clients when a media type configuration object of the specified media type is disabled, Interaction Server Proxy does not allow submissions of such interactions.

## Interaction Server Application Type

Interaction Server Proxy connects only to Interaction Servers configured with the Interaction Server application type.

## Error Codes

Interaction Server Proxy handles some Agent state information internally, and may instantly respond to client requests. New error codes, applying to specific scenarios and conditions, have been introduced because of this change of behavior. Client applications must be adjusted so as to handle these error codes. The codes are as follows:

| Error Code | Description |
|---|---|
| 2000 | Unsupported operation |
| 2001 | There are no servers accepting this media/tenant |
| 2002 | Lost connection to one of the asked servers |
| 2003 | Interaction not accessible |
| 2004 | Operation already in progress |
| 2005 | More than one server contain interaction with same ID |
| 2006 | Snapshot not accessible |

## Implementation Limitations

The proxy processes many requests by routing them to one or many Interaction Servers, based on data contained in the request. The proxy gets the tenant ID, media type, and interaction ID from the request (including ESP requests) in a generic way. Based on the values of these attributes, the proxy selects the Interaction Server.

In the following cases, Interaction Server Proxy (not Interaction Server) generates an error message:

- One of the attributes required by the proxy to route some specific request is not present.
- An attribute is provided with the incorrect type.
- An attribute is provided with an incorrect value.

Most of the time, the error message is the generic `Interaction is not accessible.`

# Migrating to ODBC from DB Server (8.5.3 Only)

Starting with release 8.5.3, you must configure Interaction Server to use ODBC database connections. Support for DB Server is deprecated and existing environments must migrate to using ODBC connections.

This page explains how to migrate existing DB Server connections to use ODBC connections.

> **Important**
> This page does not provide complete, step-by-step instructions. You must understand such concepts as DSN, ODBC, basing database configuration, and ODBC driver configuration before proceeding with migration.

### Important note

Some technical details on the following pages are obsolete and do not apply to Interaction Server 8.5.3:

- Deploying Interaction Server in this guide
- ODBC Connection in this guide
- ODBC Drivers in the eServices Integrated Capture Points Guide

In particular:

- Support of 32-bit platforms was discontinued in the 8.1.3 release. Interaction Server requires a 64-bit operating system and a 64-bit version of the ODBC driver.
- Support of the IBM DB2 database was discontinued in the 8.5.2 release.
- Support of the Solaris and IBM AIX platforms was discontinued in the 8.5.3 release.
- Database vendors have released new ODBC drivers with simplified installation processes.

## Installing ODBC drivers

> **Important**

- As Interaction Server is a 64-bit application, the ODBC driver must support 64-bit architecture.
- Installation of ODBC drivers might require administrator-level account privileges.
- Interaction Server does not support alternative ODBC drivers and tools from third-party brands, unless they are explicitly listed in this document.

You must ensure that ODBC drivers are installed on each Interaction Server machine in your environment (primary and backup).

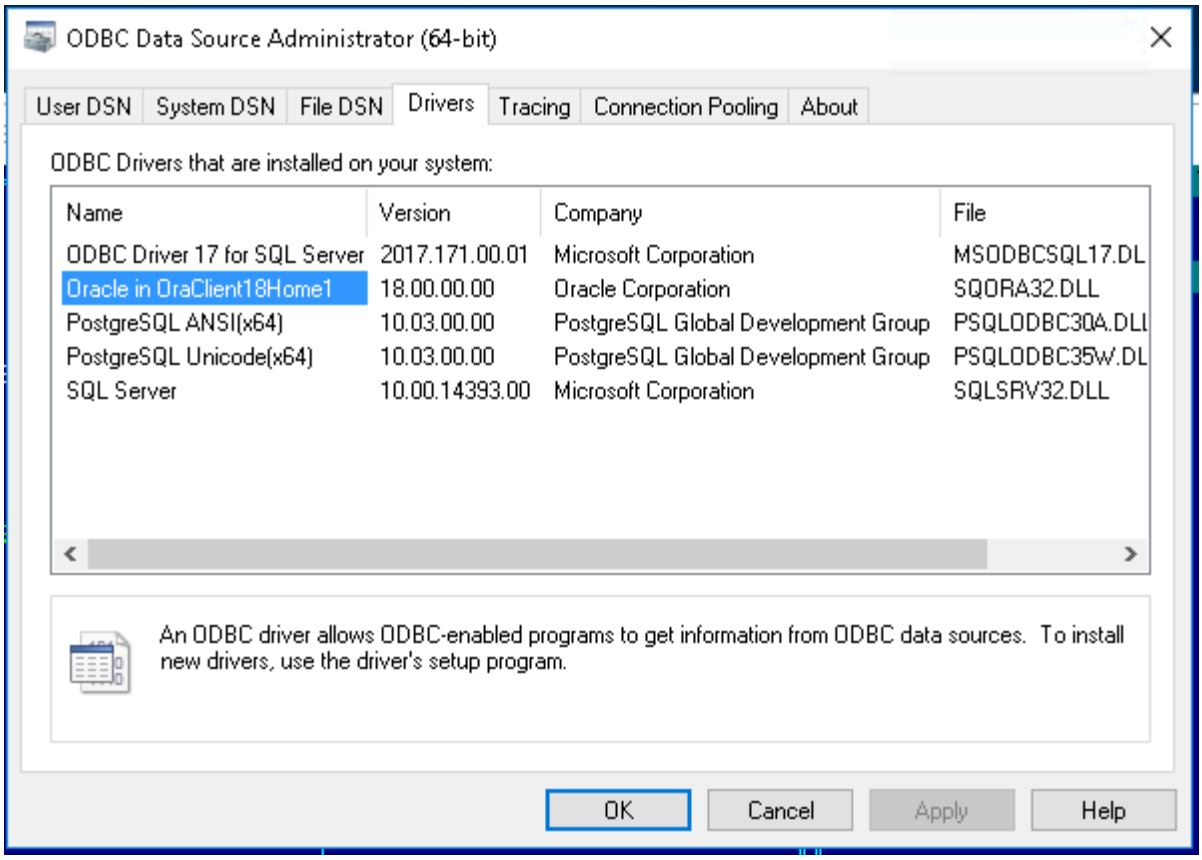
- For Microsoft SQL Server, Genesys recommends the most recent 17.x driver release. See one of the following pages for more information: Windows or Linux.
- For Oracle, Genesys recommends the most recent 18.x driver release. See one of the following pages for more information: Windows or Linux.
- For PostgreSQL, Genesys recommends the most recent 10.x driver release. See one of the following pages for more information: Windows or Linux.

It is recommended to configure the Data Source Name (DSN) for the hosts on which Interaction Server will run (see the DSN article in Wikipedia for more information). Follow the vendor instructions to install the ODBC driver and configure the DSN. Verify a successful connection to database with the tools that accompany the driver.

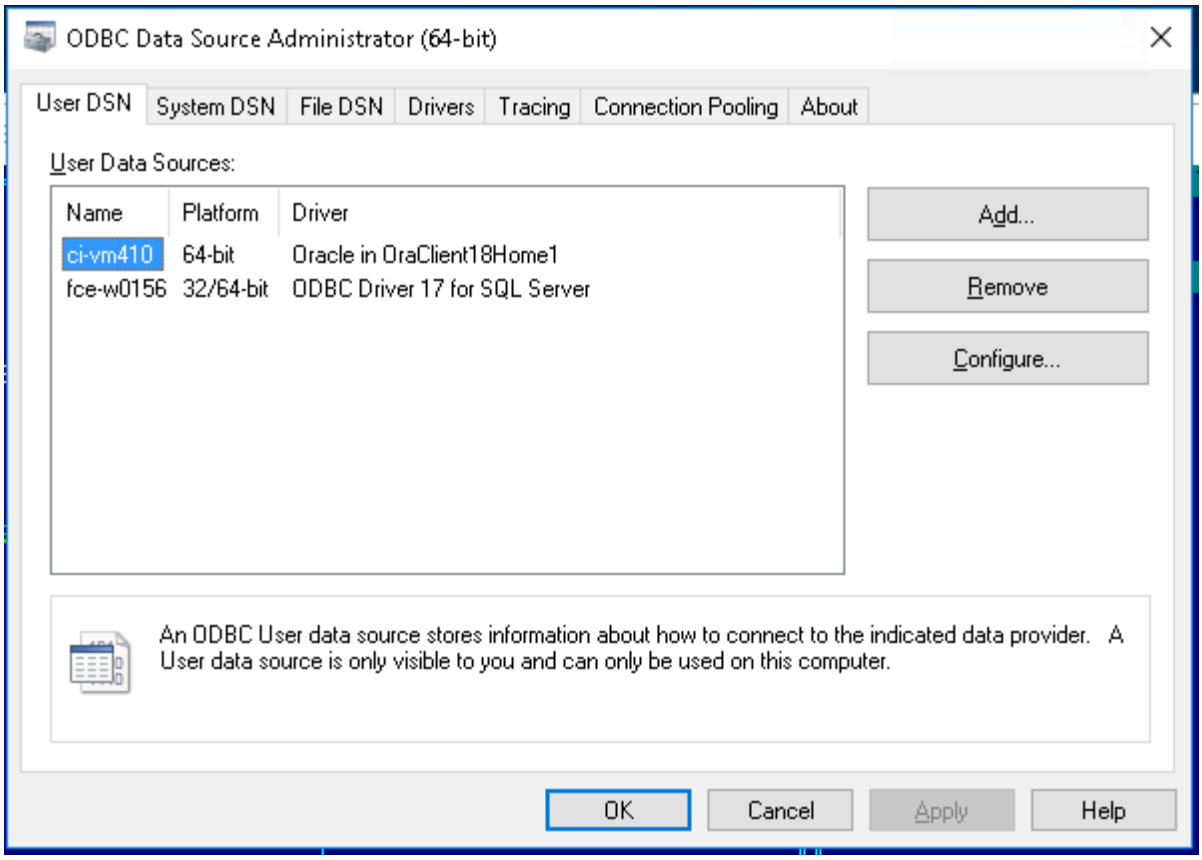
## Configuring the DSN

You can use the **ODBC Data Source Administrator** in Windows to create driver-specific DSN objects:

1. Open a command prompt.
2. Enter the command `odbcad32` to open the **ODBC Data Source Administrator**.
3. Select the **Drivers** tab to view the ODBC drivers installed in your environment. Note the names of the available drivers.

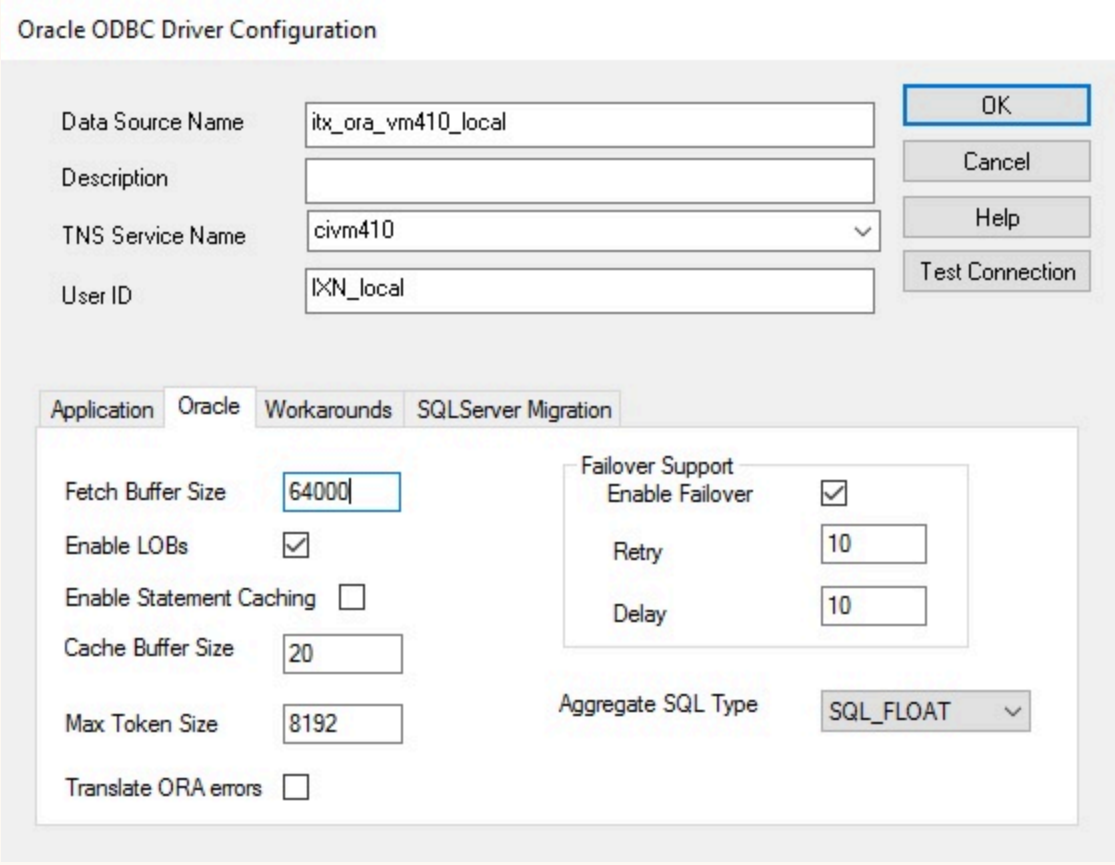



4. Switch to the tab for DSN creation:
   - If Interaction Server is started as a console application or by LCA (Local Control Agent), you must use the tab **User DSN** .
   - If Interaction Server is started as a service by the SYSTEM account on Windows, you must use the tab **System DSN**.




5. Select **Add** and follow the vendor instructions to configure the DSN, specific to your database engine and the selected driver.

> **Important**
>
> For Oracle, ensure you check **Enable LOBs** in the **Oracle** tab.




## Preparing the database

Before starting the migration, you must ensure the Interaction Server database uses the latest version of the database schema.

For Oracle databases, you must ensure that the field **flexible_properties** is of datatype **BLOB**. You must also convert your old data, if necessary.

### Using the BLOB datatype on Linux

If you are using Linux, add the line `Lobs=T` in the **odbc.ini** file configured for the Interaction Server host.

### Converting existing data

Perform the following steps to convert the **flexible_properties** column to the **BLOB** datatype:

1. Stop Interaction Server.
2. Convert the **flexible_properties** column datatype by executing the following script:

```
alter table interactions modify (flexible_properties blob default null);
```

3. Start Interaction Server.

> **Important**
>
> You must ensure that the preconditions mentioned in the Oracle documentation are met. Databases that are created with default scripts from Genesys should meet these preconditions, but it's recommend that customers also verify these preconditions are met.

## Preparing DAP configuration objects

### New environment

For a new environment, create a Database Access Point (DAP) object as described in the Framework Database Connectivity Reference Guide and add the DAP to the connections of your Interaction Server.

To configure the database-oriented Event Logger for Interaction Server, create a second DAP object as described on the Deploying Event Logger page.

### Existing environment

For an existing environment, use Genesys Administration Extension (GAX) to create copies of the existing DAP objects used by Interaction Server. You can use these copies to switch back to connecting via DB Server while testing.

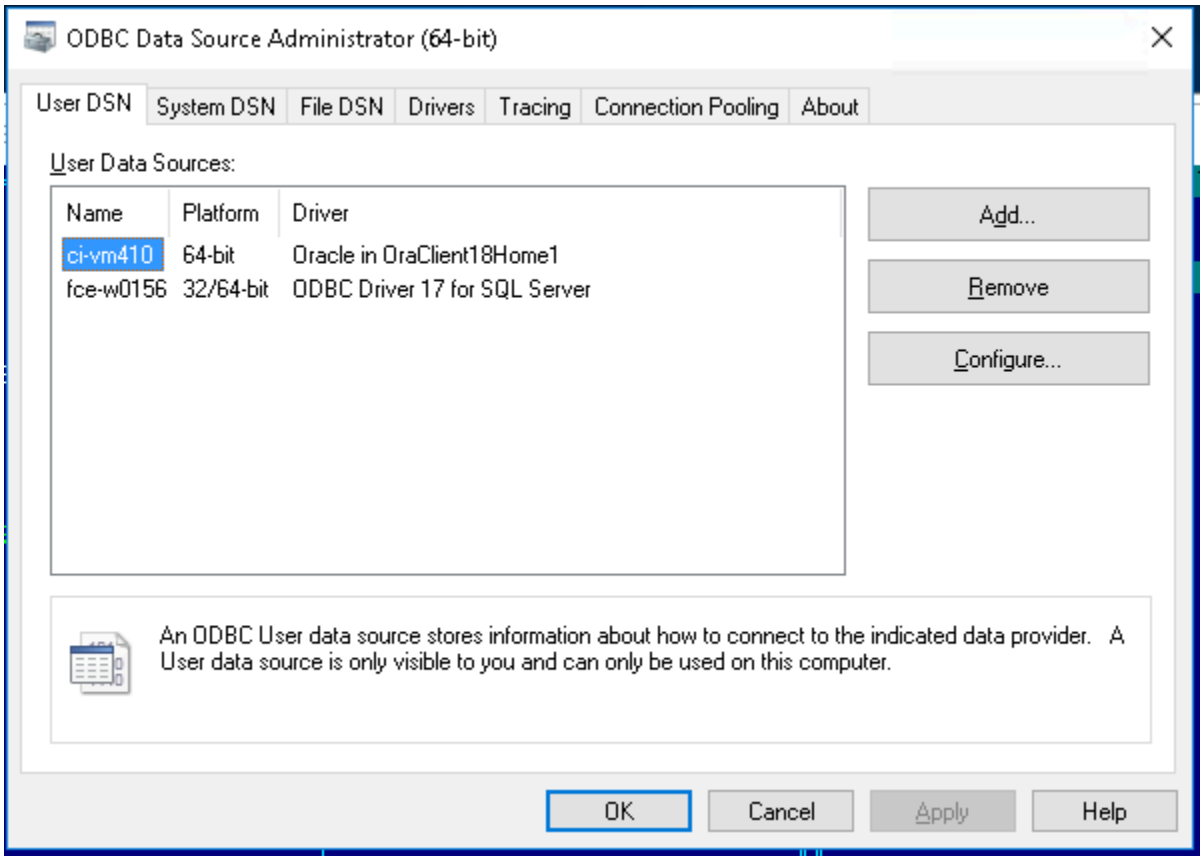
## Updating configuration objects to use ODBC

In the DAP object that defines the connection to the Interaction Server main database:

1. Create or change the option dbprotocol with the value `odbc`.
2. Create or change the option odbc-string to one of the following values:
   - If you have created a DSN object for the main database, enter the value `"DSN=name_of_DSN"`.
   - Otherwise, enter `"Driver={name_of_driver}"`.

In the sample picture below, the respective values are:

- `DSN=ci-vm410`
- `Driver={ODBC Driver 17 for SQL Server}`




For the main database, all options should reside in the **settings** section. For the event log database, all options should reside in the **logger-settings** section.

The steps are similar to set up the DAP object for the Event Logger database:

1. Create or change the option delivery-protocol to the value `odbc`.
2. Create or change the option odbc-string to one of the following values:
   - If you have created a DSN object for the Event Logger database, enter the value `"DSN=name_of_DSN_for_Logger"`.
   - Otherwise, enter `"Driver={name_of_driver}"`.

If you are using the same location for the main database and the Event Logger database, the values of these options might be identical. However, for performance or other reasons, you might have the Interaction Server main database and the Event Logger database installed in different locations.

> **Important**
>
> - Genesys recommends that you do not use different database engine types simultaneously. For example, do not use Oracle for the main database and Microsoft SQL Server for the Event Logger database.
> - Do not use curly brackets ({ or }) around DSN names. Curly brackets can only be used around driver names.

## Additional information for Linux

If Interaction Server is running on Linux, the value for the option odbc-string in DAP objects might depend on where the driver's files were installed. For example, if you are not using a DSN, the driver name might be the actual location of the file, such as in the examples below:

- Microsoft SQL Server: `Driver=/opt/microsoft/msodbcsql17/lib64/libmsodbcsql-17.2.so.0.1`
- Oracle: `Driver=/usr/lib/oracle/18.3/client64/lib/libsqora.so.18.1`
- PostgreSQL: `Driver=/home/username/pgodbc/psqlodbc-version/lib/psqlodbcw.so`

To use a DSN, Linux users can install the ODBC manager unixODBC, an open-source third-party application. Database and ODBC driver vendors state information about compatibility between unixODBC and driver versions in their installation instructions.

> **Tip**
>
> Refer to the unixODBC User Manual for instructions on how to test an ODBC connection using the **isql** tool.

## Testing the ODBC connection with Interaction Server

Perform the following steps to test the ODBC connection:

1. Configure Interaction Server to collect the log into a text file with debug-level details. See the Options Reference for more information on available options.
2. Start Interaction Server and let it run to generate some events for the log.
3. Stop Interaction Server.
4. Open the log and review all parts that include the string **odbc** or the names of associated DAP objects. In the case of a successful ODBC connection, the log contains the following messages:
    - Connection to main database:

(before 8.5.303.08)

```
Std 27126 New database connection opened (connection ID: 1)
 Std 27113 Checking database integrity (connection ID: 1)
```

(since 8.5.303.08)

```
Std 27128 New database connection opened (connection ID: 1, host: OracleDB)
 Std 27113 Checking database integrity (connection ID: 1)
```

5. Connection to Event Logger database:

```
Std 27126 New database connection opened (connection ID: 100001)
```

> **Important**
>
> - The connection ID number **100001** indicates a connection to the Event Logger database.
> - Interaction Server might open more than one connection to the database when necessary and as directed by the values of the option number-of-database-connections. Subsequent connection ID numbers are incremented accordingly.
> - The number of database connections to the Event Logger database can be specified in the **number-of-database-connections** option in the **setting** section of the Event Logger.

## Troubleshooting a failed connection

See below for quick troubleshooting hints for first-time connections. Make the necessary changes on the database engine side or in the DAP object.

### All databases

- On Linux, you might need to have the location of the ODBC driver listed in the environment variable **LD_LIBRARY_PATH**. This also applies to the unixODBC libraries location.

### Microsoft SQL Server

- Ensure the instance name is correct.

### Oracle

- Ensure the SID or ServiceName values are correct, and check the listener resolution in the file **tnsnames.ora**.
- Ensure the setting of the environment variable **TNS_ADMIN** points to the folder where the file **tnsnames.ora** is located.
- On Linux, the driver might require the following environment variables to be set:
  ```
  export NLS_NUMERIC_CHARACTERS=""
  ```

```
export NLS_LANG=".UTF8"
export ORACLE_SID=<ORCL_actual_name>
```

### PostgreSQL

- Ensure that permissions to connect from a remote host are managed in the configuration file(s) **pg_hba.conf**.