



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

GIS Deployment Guide

Genesys Interaction SDK 7.6.6

Table of Contents

Genesys Integration Server Deployment Guide	3
Table of Contents	6
Change History	8
Introducing GIS	14
Task and their Related Procedures	30
Installing and Uninstalling	34
Customizing the GIS Environment	49
High-Availability Deployments of GIS	60
Starting and Testing GIS	76
Configuration Options	85
Agent Interaction Service Options	86
Open Media Service Options	96
Genesys Integration Server Options	100

Genesys Integration Server Deployment Guide

Welcome to the *Genesys Integration Server Deployment Guide*. This document introduces you to the concepts, terminology, and procedures relevant to the Genesys Integration Server 7.6. Topics covered in this guide include:

- Genesys Integration Server (GIS) architecture
- Deployment-planning considerations
- GIS installation and configuration procedures
- Instructions for starting and stopping GIS
- Instructions for generating the WSDL files for the Session, Statistics, and Configuration services

Warning

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

About Genesys Integration Server

Genesys Integration Server (GIS) provides an interface to the Genesys Framework—specifically to Stat Server statistics, Configuration Database configuration information, the Agent Interaction services, and the Open Media Interaction services. It presents APIs that enable you to create client applications designed to access and modify Genesys data, or to build agent-facing applications based on your own customized, web-based applications.

Intended Audience

This document, primarily intended for managers who will be planning the deployment of GIS, system administrators, and integrators. It assumes that you have a basic understanding of:

- Computer-telephony integration (CTI) concepts, processes, terminology, and applications.
- Network design and operation.
- Your own network configurations.

You should also be familiar with the architecture and functions of the Genesys Framework and Customer Interaction Management Platform—especially with the Configuration Layer and with Stat Server.

Chapter Summaries

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

- **Introducing GIS**, presents an overview of GIS functionality and architecture, plus deployment-planning information.
- **Installing and Uninstalling GIS**, provides step-by-step instructions for installing, uninstalling, and deploying GIS.
- **Customizing the GIS Environment**, provides options for reconfiguring GIS after installation.
- **High-Availability Deployments of GIS**, provides details on how to deploy GIS in high-availability (HA) mode, and describes the load balancing capabilities of those deployments.
- **Starting and Testing GIS**, provides step-by-step instructions for starting and stopping GIS and testing your installation.
- **Configuration Options**, configuration information for the options tab.

List of Procedures

- **Installing GIS on a UNIX-Based Platform**
- **Installing GIS on a Windows Platform**
- **Deploying GIS in WebLogic**
- **Deploying GIS in WebSphere**
- **Inspecting the Stand-alone Directory Tree**
- **Inspecting the WebSphere Directory Tree**
- **Inspecting the WebLogic Directory Tree**
- **Checking the GIS Log Files**
- **Replacing the JavaMail Archive (for WebSphere 5.0)**
- **Uninstalling GIS as a SOAP Stand-alone Server from a Windows Platform**
- **Uninstalling GIS as a SOAP Stand-alone Server from a UNIX-Based Platform**
- **Uninstalling GIS from a WebSphere Application Server**
- **Uninstalling GIS from a WebLogic Application Server**
- **Setting the JAVA_HOME Environment Variable**
- **Tuning Heap Size Values**
- **Configuring a SOAP Port for a Master Configuration Server**

- [Configuring a SOAP Port for a CS Proxy](#)
- [Customizing the Notification Mechanism for GIS:SOAP](#)
- [Setting the HTTP Connector Thread Number for GIS:SOAP](#)
- [Displaying the WSDL Files for GIS:SOAP](#)
- [Enabling GZIP Compression for .NET](#)
- [Enabling GZIP Compression for Java](#)
- [Enabling GZIP Compression using Web Server Features](#)
- [Configuring the GIS Application Objects](#)
- [Configuring the HA Cluster for Manual Peer Discovery Support](#)
- [Configuring the HA Cluster for Automatic Peer Discovery Support](#)
- [Configuring the Client Application](#)
- [Configuring Apache Dispatcher](#)
- [Configuring Tomcat for Apache Dispatcher Deployment](#)
- [Configuring and Installing GIS](#)
- [Configuring the Server Side](#)
- [Configuring the Client Side with Dispatcher](#)
- [Configuring the Client Side without Dispatcher](#)
- [Choosing Your Start Script](#)
- [Editing the Start Script](#)
- [Starting and Testing GIS](#)
- [Configuring the License Section of the Option Tab \(for GIS:SOAP\)](#)
- [Specifying Licensing at Startup \(GIS:SOAP only\)](#)
- [Starting on UNIX-based Platforms \(GIS:SOAP only\)](#)
- [Verify the Status of GIS \(GIS:SOAP Only\)](#)
- [Starting GIS on Windows](#)
- [Starting GIS on a Web Application Server](#)
- [Stopping GIS on UNIX-based Platforms](#)
- [Stopping GIS on Windows](#)
- [Stopping on a Web Application Server](#)
- [Using Management Layer to Start and Stop GIS](#)

Table of Contents

Genesys Integration Server Deployment Guide

[About Genesys Integration Server](#)
[Intended Audience](#)
[Chapter Summaries](#)
[List of Procedures](#)
[Related Resources](#)

Introducing GIS

[About the Genesys Integration Server](#)
[GIS Architecture](#)
[System Requirements](#)
[GIS Use-Case Scenario](#)
[Licensing](#)
[New In This Release](#)

Installing and Uninstalling GIS

[GIS Installation and Deployment Options](#)

Customizing the GIS Environment

[Windows Service Setting](#)
[LCA Configuration](#)
[Tuning the Heap Size Values](#)

Customizing the GIS Environment (more)

[SDK Server Connections for GIS](#)
[SOAP, Configure the Configuration Server](#)
[SOAP Port](#)
[Customize the Notification Mechanism,](#)
[The HTTP Connection](#)

High-Availability Deployments of GIS

[Overview](#)
[Deployment for High Availability \(No](#)
[Load Balancing\)](#)
[Deployment for High Availability with](#)
[Load Balancing](#)

Starting and Testing GIS

Starting GIS

Stopping GIS

Agent Interaction Service Options

dn-at-switch Section

kworker Section

license Section, loading Section, log
Section

multimedia Section

outbound Section

Open Media Interaction Service Options

esp Section

general Section

log Section

server Section, settings Section

ucs Section

workflow Section

Genesys Integration Server Options

ail-services Section

core-services Section

license Section, log Section

session-services Section, stat-services
Section

Change History

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

Version 7.6.609.00

Page name	State	Additional details
Introduction	Updated	<ul style="list-style-type: none">• Updated sections<ul style="list-style-type: none">• Container and Client-Server Options• GIS Services as Represented in the Configuration Layer• Configuration SDK Service• Statistics SDK Service• Agent Interaction Services• GIS Architecture• GIS Built with Flexible Technologies• Supported Toolkits• System Requirements• Supported Operating Systems• Framework and Solutions Compatibility• Statistics SDK Service• Licensing
InstallingandUninstalling	Updated	<ul style="list-style-type: none">• New sections<ul style="list-style-type: none">• Uninstalling GIS as a SOAP Stand-alone Server from a Windows Platform• Uninstalling GIS as a SOAP Stand-alone Server from a UNIX-Based Platform• Updated sections

Page name	State	Additional details
		<ul style="list-style-type: none"> • GIS Installation and Deployment Options • Installing GIS on a UNIX-Based Platform • Installing GIS on a Windows Platform • Deploying GIS in WebSphere • Inspecting the WebSphere Directory Tree • Deleted sections <ul style="list-style-type: none"> • Verifying Installation of GIS as a GSAP Server • Replacing the JavaMail Archive (for WebSphere 5.0) • Uninstalling GIS as a GSAP or SOAP Stand-alone Server from a Windows Platform • Uninstalling GIS as a GSAP or SOAP Stand-alone Server from a UNIX-Based Platform
Customizing the GIS Environment	Updated	<ul style="list-style-type: none"> • Updated sections <ul style="list-style-type: none"> • Windows Service Setting • Tuning Heap Size Values • Deleted sections <ul style="list-style-type: none"> • Load Balancing Under GSAP • GSAP Load Balancing Configuration
High-Availability Deployments	Updated	<ul style="list-style-type: none"> • Updated sections <ul style="list-style-type: none"> • Overview • Configuring the HA Cluster for Manual Peer Discovery Support

Page name	State	Additional details
		<ul style="list-style-type: none"> • Configuring the HA Cluster for Automatic Peer Discovery Support • Deployment for High Availability with Load Balancing • Alternative Deployment for High Availability • Deleted sections <ul style="list-style-type: none"> • GSAP High Availability Limitations
StartingandTestingGIS	Updated	<ul style="list-style-type: none"> • New sections <ul style="list-style-type: none"> • Configuring the License Section of the Option Tab for GIS SOAP • Specifying Licensing at Startup (GIS SOAP) • Starting on UNIX-based Platforms (GIS SOAP) • Verify the Status of GIS (GIS SOAP) • Updated sections <ul style="list-style-type: none"> • Choosing Your Start Script • Editing the Start Script • Starting and Testing GIS • Starting GIS on Windows • Stopping GIS on UNIX-based Platforms • Stopping GIS on Windows • Deleted sections <ul style="list-style-type: none"> • Configuring the License Section of the Option Tab (for GIS SOAP or GIS GSAP) • Specifying Licensing at Startup (GIS SOAP only) • Starting on UNIX-based Platforms (GIS SOAP only)

Page name	State	Additional details
		<ul style="list-style-type: none">• Verify the Status of GIS (GIS SOAP Only)
GenesysIntegrationServerOptions	Updated	<ul style="list-style-type: none">• New sections<ul style="list-style-type: none">• gis_interactionservice-firstblock-size• gis_interactionservice-maxblock-size• gis_stat-service-firstblock-size• gis_stat-service-maxblock-size• gis_configuration-service-firstblock-size• gis_configuration-service-maxblock-size• multimedia Section• logout-on-last-media• Updated sections<ul style="list-style-type: none">• gis_interactionservice-block-size• gis_configuration-service-block-size• log Section• all• Buffering• debug• expire• MessageFile• segment• standard• trace• verbose• Deleted sections<ul style="list-style-type: none">• lb-console• lb-file

Page name	State	Additional details
		<ul style="list-style-type: none"> • lb-msgsrv • prop-console • prop-file • prop-msgsrv • [GSAP.General] Section • ha-session-ttl • max.param.count • nat-mode • server.backlog • server.clientsocket.enablekeepalive • server.clientsocket.enabletcpnodelay • server.clientsocket.sendbufsize • server.clientsocket.sotimeout • server.handlers.max • server.handlers.recommended • [GSAP.KeepAlive] Section • handler-daemon-audit-interval • handler-daemon-inactivity-interval • ping-daemon-audit-interval • ping-daemon-inactivity-interval • ping-times-threshold • session-timeout-threshold • [GSAP.LoadBalancing] Section • lb-clientsocket-enabletcpnodelay • lb-clientsocket-sotimeout • lb-cluster-servers • lb-connector-exchange-interval • lb-connector-process-

Page name	State	Additional details
		<p>interval</p> <ul style="list-style-type: none">• lb-load-threshold• lb-load-delta-threshold• lb-max-connections-per-server• lb-max-transfer-count• lb-port• lb-process-interval

Introducing GIS

This chapter describes Genesys Integration Server (GIS) architecture, the APIs and services that GIS exposes, and the basic requirements for deployment.

About the Genesys Integration Server

GIS is an XML server implemented as a web application. Developers use GIS to access the Genesys Framework and the Customer Interaction Management Platform by using one or more APIs to expose available SDK services.

Container and Client-Server Options

GIS is an XML server implemented as a web application that is embedded in a Tomcat, WebSphere, or WebLogic web container.

Important

This guide refers to the Tomcat deployment mode as a "stand-alone deployment." GIS 7.6 supports Tomcat versions 5.0.x and 5.5.x.

GIS is compatible with the Java 7 Platform, Enterprise Edition (J2EE). GIS is based on open Internet technologies and industry standards, such as eXtensible Markup Language (XML) and Simple Object Access Protocol (SOAP).

Important

In 7.6.6, GSAP protocol is discontinued

The GIS client interface conforms to Web Services Description Language (WSDL) specifications. It uses HTTP as the underlying message transport layer. Because GIS is based on common technologies, you can develop your client applications using one of several supported toolkits. For a list, see [GIS Built with Flexible Technologies](#).

For details, see the [Genesys Supported Operating Environment Reference Guide](#).

GIS Services as Represented in the Configuration Layer

To integrate available GIS services into the Configuration Layer of a Genesys environment, you need to map those services to Genesys Configuration Layer Application objects. The services then become Configuration Layer entries that expose Genesys Integration Server services. Services do not function

as servers. Each instance of Agent Interaction Service provides you with connections to the T-Servers, Interaction Server, and Universal Contact Server and each instance of Open Media Interaction Service provides you with connections to Interaction Service and Universal Contact Server.

Each GIS-exposed Genesys SDK Service is composed of three components:

- GIS, which presents the necessary APIs.
- Documentation that explains how to develop a client application that can successfully access the API. The documentation set for each SDK Service includes a *Developer's Guide* and an *API Reference*. The developer's guides also contain the code used in the code examples, accompanied by brief comments explaining the methods used, for instance, the Agent Interaction SDK Services Developer's Guide.
- Code examples, available on the [ISDK code samples page](#), consisting of functioning code examples in Java and C# that *exercise-related functionality and provide a model for how to write code to make your application perform specific functions*.

Important

All code examples are as accurate as possible. However, they are examples and are not tested or supported as fully functional code. Genesys does not guarantee that you can join the examples to create a complete application.

Session Service

Session Service—provides login, logout, and licensing functionality.

The Session Service provides an interface for login, logout, and licensing functions. Your client application must address the Session Service and receive licensing validation before it can communicate with the other services. The Session Service does not include a documentation component.

Configuration SDK Service

Configuration Service—provides an interface to Configuration Layer objects, enabling your client applications to access and modify Genesys configuration data.

The Configuration SDK Service enables you to view and modify configuration information in the Genesys Configuration Layer, using a locally developed, user-customized application.

Possible customizations include the following scenarios:

- A multi-site enterprise might provide access to configuration data for every tenant.
- Your client application might present data to a web page, which off-site supervisors and administrators can access.
- Your application might retrieve configuration updates based on time or change criteria.
- You might configure client applications or their users to access subsets of data, for instance, configuration data for only certain switches, queues, or agent groups.

- Your enterprise might use a centralized client application to retrieve, by way of the Internet, updated configuration information from multiple GIS servers deployed at different sites.
- Your third-party application can be integrated with Genesys Configuration to provide real-time synchronization, in one or both directions.

Examples of some of the above functionality are reproduced, with comments and suggestions for developing your client applications, in the Configuration SDK Web Services Developer's Guide.

Statistics SDK Service

Statistics Service—provides an interface to Stat Server statistics, granting your custom client applications access to Genesys real-time and historical data.

The Statistics SDK Service facilitates development of client applications that gather statistics about resource utilization, while introducing tools, such as filters, that enable a business-driven evaluation of contact-center operation and agent performance.

Users with a license only for the Statistics SDK Service can access read-only configuration information through the Configuration SDK Service using a limited selection of the Configuration Service methods.

Client applications using the Statistics SDK Service might perform such specialized functions as the following:

- In an enterprise with many T-Servers® and Stat Servers, provide access to statistical subsets for particular T-Servers, switches, or agent groups.
- Present data to a desktop application, to a wallboard display, or to a particular web page.
- Retrieve statistics on contact-center traffic, or employee productivity, based on time or change criteria.
- Access subsets of data, for instance, data that helps supervisors monitor agent performance or that enables managers to monitor campaign effectiveness.
- Retrieve, using the Internet, updated statistical information from GISs at multiple sites.

Examples of some of the above functionality are reproduced, with comments and suggestions for developing your client applications, in the Statistics SDK Web Services Developer's Guide.

Agent Interaction Services

Agent Interaction services—facilitates development of custom applications for interaction handling. For example, voice, e-mail, and/or chat interactions.

The Agent Interaction services enable you to develop applications for purposes like the following:

- Create a contact-center agent desktop application for Genesys software implementations.
- Integrate Genesys software with third-party software.
- Create other, specialized applications tailored to your needs. Typical usage scenarios include:
 - Managing agent login activity.
 - Handling e-mail interactions: sending, receiving, replying.
 - Handling voice interactions: calling, receiving, callback.
 - Handling chat interactions.

- Handling third-party media interactions.
- Handling outbound campaign participation.

Examples of some of the above functionality are reproduced, with comments and suggestions for developing your client applications, in the Agent Interaction SDK Java Developer's Guide. Typically, your applications will use the Agent Interaction services to establish connections to Genesys' Framework, Internet Contact Solution, and Outbound Contact Solution servers; to manage agent login activities; and to manage interactions.

Important

- Route Point and Route Service functionality are restricted within the Agent Interaction services, as deployed on GIS.
- Since 7.6.4, Agent Interaction services supports only one Stat Server connection

Open Media Interaction Services

Open Media Interaction services—enable you to build applications that manage third-party media interactions in the Genesys Framework. Such applications might submit third-party media interactions to Interaction Server; handle interactions' extensions through Interaction Server, using the External Service Protocol (ESP); manage third-party media interactions in the Universal Contact Server database; or manage the system's Customer Interaction Management queues. The Open Media Interaction services enable you to develop applications for purposes like the following:

- Maintaining a queued interface for Genesys media types and for third-party media.
- Getting business attributes and their values.
- Monitoring the changes in queues' state, and in associated interactions.
- Getting events on interactions in queues.
- Maintaining a media interface for third-party media types.
- Creating and managing third-party media interactions submitted to Interaction Server.
- Managing third-party media interactions in the Universal Contact Server database.
- Using the External Service Protocol (ESP) to handle interactions' extensions, through Interaction Server.

Examples of some of the above functionality are reproduced, with comments and suggestions for developing your client applications, in the Open Media Interaction SDK Services Developer's Guide.

GIS Architecture

The following architecture diagram shows the connections between GIS, the Genesys Framework, and your client application.

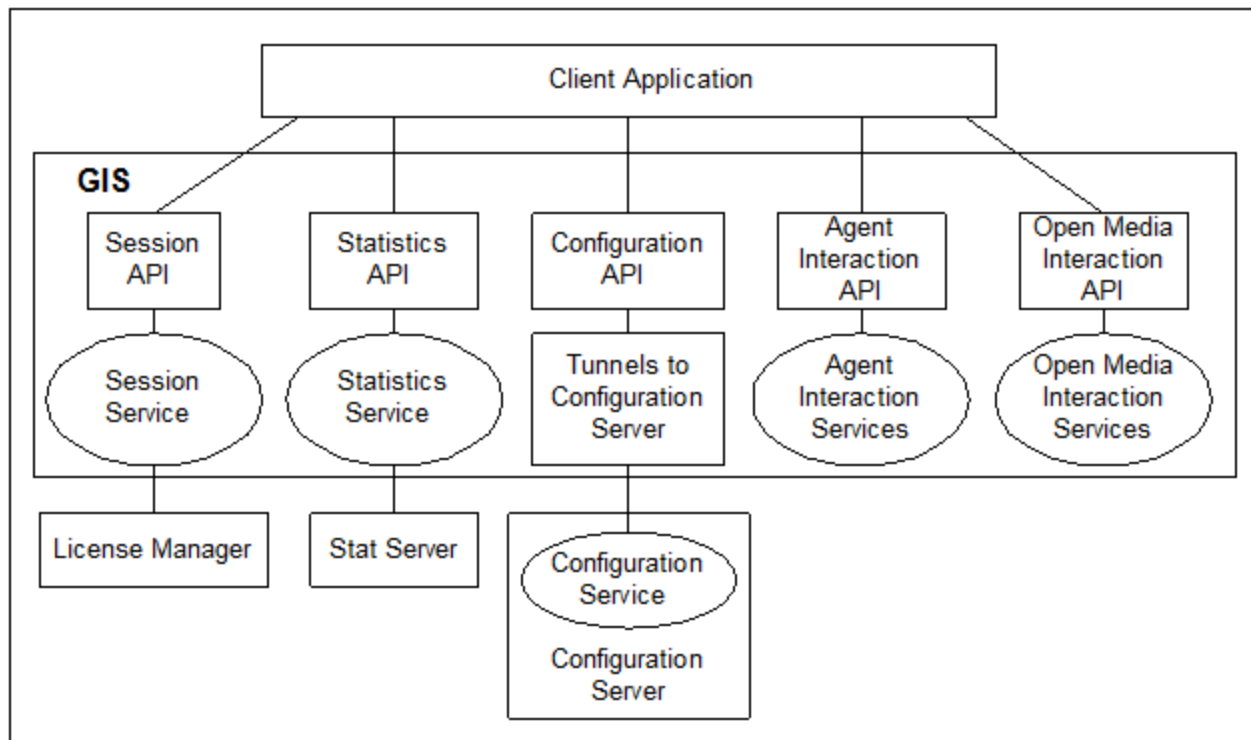
Important

- All the APIs and services shown in the diagram below]] are available to GIS:SOAP installations. Those labeled with an asterisk (*) are available only to GIS:SOAP installations.
- A single instance of GIS can support multiple client applications at the same time.

GIS works with the Genesys Framework in the following ways:

- GIS uses Configuration Server and License Manager for authentication and authorization.
- GIS accesses Configuration Server for connectivity information for Genesys Framework components.
- GIS supports integration with the Management Layer and is visible from Solution Control Interface.
- GIS communicates with Stat Server to retrieve statistical information that GIS presents through the Statistics Service.
- GIS communicates with Configuration Server's SOAP interface to access and modify configuration data that GIS presents through the Configuration Service.
- GIS communicates with the Agent Interaction Services to handle interactions on voice, e-mail, or chat media.

You can deploy single or multiple GIS instances that access the same or multiple Framework components. These connections are configured using Configuration Manager.



GIS Overall Architecture

GIS Built with Flexible Technologies

GIS client applications communicate with GIS using SOAP over HTTP only from release 7.6.6. For SOAP clients, communication conforms to a set of request/response operations that use XML as a basis for SOAP and WSDL definitions and behaviors.

If you are developing in Java, or in .NET-supported languages like Microsoft C#, Visual C++ .NET, or VB.NET, you can generate stubs or proxies that connect your application to GIS. However, Genesys recommends that you use the Genesys Integration Server Proxies/Libraries that Genesys provides (in both Java and .NET versions for the SOAP protocol). These proxies/libraries incorporate reconnect and defense mechanisms, as well as performance enhancements.

By deploying GIS as a SOAP connector, its reliance on HTTP as the underlying message transport layer means you can use existing, well-known communications and security functions in your applications. Only HTTP ports need to be opened. This permits minimal data transfer, which enables the use of standard ports and speeds up communications.

Supported Toolkits

Because of their XML-based nature and open architecture, the Statistics SDK Service and Configuration SDK Service enable developers to choose among various commercially available tools to build a SOAP-compliant client application that uses HTTP as the transport protocol. You should be able to use most major-market toolkits to develop successful client applications. The following have been tested and are officially supported:

- Microsoft .NET Framework SDK, version 2.0, 3.0, 3.5, 4.0 and 4.5 (<http://msdn.microsoft.com/>)

```
netframework/)
```

- Apache AXIS toolkit, version 1.4 (<http://xml.apache.org/axis/index.html>)

Adding a Web Server in Front of GIS

GIS resides in Tomcat, WebSphere, or WebLogic. If you plan to put an additional web server in front of GIS, see that web server's documentation for deployment instructions.

System Requirements

GIS is compatible with several hosts. For details about GIS system requirements, see the [Genesys Hardware Sizing Guide](#) . Generally, the GIS host should:

- Have at least 8 GBytes RAM.
- Have a CPU speed above 1.5 GHz, with 1.7 GHz as a minimum for Windows platforms.

Important

GIS software is multi-threaded and can take advantage of hosts with multiple CPUs.

- Have network access of at least 10 Mbps. However, Genesys highly recommends 100 Mbps or higher, in particular for SOAP deployments.

Install the Java Runtime Environment (JRE) or Java JDK, version 7 or 8:

- Windows and Solaris users should download the necessary software from the Oracle site at <http://www.oracle.com/technetwork/java/javase/downloads/index.html> .
- For Linux (Red Hat)
 - For REHL6:

Download and install Java from the Oracle site: [here](#).

Install also the 32 libs C++ package (which is not installed by default), required for the GIS installer package. To check the list of installed packages, use this command.

```
rpm -q compat-libstdc++-33
```

If the packages are not correctly installed, download and install the compat-libstdc++-33 package available [here](#)

To install the package, enter the following command at the prompt:

```
rpm -i compat-libstdc++-33-3.2.3-69.el6.i686
```

```
rpm -i compat-libstdc++-33-3.2.3-69.el6.x86_64
```

- For REHL7, check that the compatibility packages are installed:

```
rpm -q compat-libstdc++-33
```

If not, install them with the following procedure:

- Enable optional repo for yum:

```
yum-config-manager --enable rhel-7-server-optional-rpms
```

- Then install the following packages:

```
yum install compat-libstdc++-33-3.2.3-72.el7.i686.rpm
```

```
yum install compat-libstdc++-33-3.2.3-72.el7.x86_64
```

- AIX users should download the necessary software from <http://www.ibm.com/developerworks/java/jdk/aix/service.html>
- HP-UX users should download the necessary software from <https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HPUXJAVAHOME>

To start GIS installation, run the `install.sh` script.

Important

Certain deployment scenarios require specific Java components:

- When deploying GIS using the Web Module option, install the full Java SDK, not just the JRE.
- When deploying GIS on the Solaris operating system, install JDK 7

For details, see the [Genesys Supported Operating Environment Reference Guide](#). Other deployment notes:

- The GIS installation package includes the recommended version of Apache AXIS and Tomcat. Genesys encourages you to use these versions to generate proxies (Tomcat 6 is supported).
- You can deploy GIS into a WebSphere or WebLogic web container.

Warning

In this release, deployment into WebLogic is supported on the Solaris platform only.

- The GIS directory tree consumes approximately 60 MB for SOAP deployments, with slight variations depending upon operating system.
- To integrate a stand-alone GIS instance with the Genesys Management Layer's Solution Control Server (SCS), you must install Local Control Agent (LCA), version 7.0.100.05 or higher, on the host that supports GIS.

Supported Operating Systems

For supported operating systems, see [Genesys Supported Operating Environment Reference Guide](#). This guide is updated regularly.

Important

Before installing GIS on a Red Hat Enterprise Linux 6 64-bit operating system, you must first install the Red Hat compatibility packages.

Sizing Guidelines

For sizing guidelines, refer to the Genesys Integration Server chapter in the [Genesys Hardware Sizing Guide](#), which you can access from the Technical Support website. This document is updated regularly.

Production vs. Development/Test Environments

You can deploy a simplified development/test environment using one of the following architectures:

- Install all required software on a single host. This includes GIS, DB Server, Configuration Server, Configuration Manager, and also all of the client development tools, including compilers, toolkits, and so on.
- Install all of the client development tools on one host and put all of the servers on another.
- Locate the client development tools and GIS on a single host, with the Genesys Framework resources on another host.

For more information, refer to the Genesys Integration Server chapter in the [Genesys Hardware Sizing Guide](#)

Under any deployment architecture, if you require the Statistics SDK Service, ensure that GIS has access to Stat Server.

Framework and Solutions Compatibility

Important

This section describes how some Genesys systems work together (interoperability).

For details, see the [Genesys Interoperability Guide](#).

Statistics SDK Service

The Genesys Statistics SDK Service is compatible with single-tenant or multi tenant Framework

versions 6.5 and higher. To manage configuration data, you must use either Configuration Server 7.x (which can be configured either as a master Configuration Server or as a proxy), or CS Proxy 6.5.

Configuration SDK Service

The Configuration SDK Service requires either Configuration Server 7.x (which can be configured as either a master Configuration Server or a proxy) or CS Proxy 6.5. The Configuration SDK Service is compatible with single tenant or multi-tenant versions of the Genesys Framework.

Agent Interaction Service

The Agent Interaction SOA (Service Oriented Architecture) is compatible with Framework 7.0 and with the following Genesys products:

- Outbound Contact Solution (OCS): 7.x
- Voice CallBack: 7.x
- Multimedia or Multi-Channel Routing (MCR): 7.x

Important

For basic Open Media features, you must have Multi-Channel Routing 7.1.x, or Multimedia 7.2.x or higher. For complete Open Media features (including Queued Interaction Layer and Media Interaction Layer), Multimedia 7.2.x or higher is mandatory.

Open Media Interaction Service

The Open Media Interaction Service is compatible with Framework 7.1 or higher, and with Multi-Channel Routing 7.1.x or Multimedia 7.2.x and higher.

Management Layer Support

If you are using GIS in a Tomcat web container, you can start and stop GIS using Solution Control Interface (SCI).

To use the Management Layer, you must be running LCA (Local Control Agent) 7.0.1 or higher.

Important

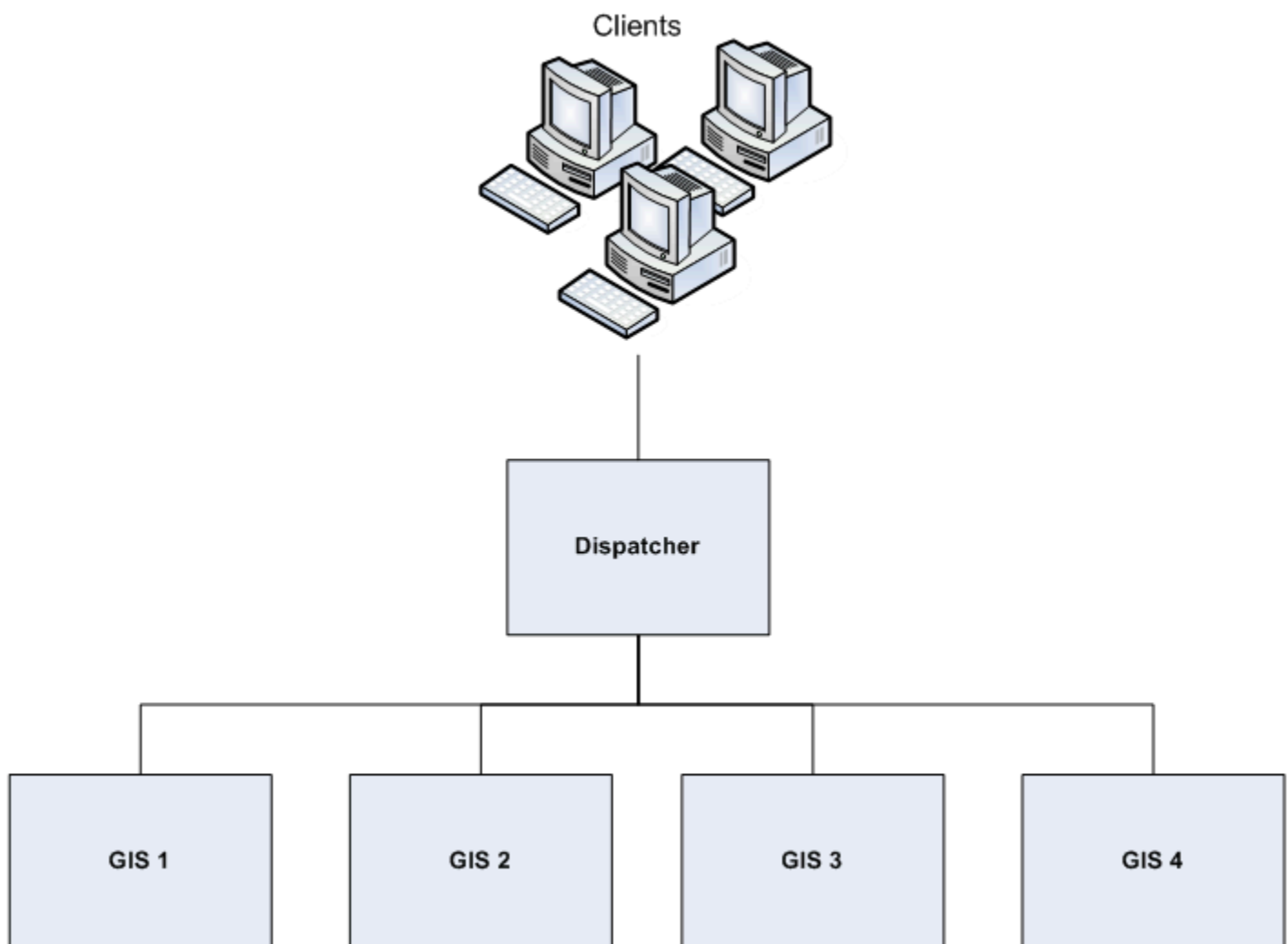
You cannot use the Management Layer to control GIS if you are running GIS in a WebSphere or WebLogic container.

GIS Use-Case Scenario

This section provides details from one GIS deployment that Genesys implemented at a customer site. The design offers a highly available, robust, and redundant architecture for two-thousand agents.

GIS:SOAP 7.6 - Clustering

The GIS:SOAP 7.6 HA with Load Balancing deployment is suitable for a highly available, robust, redundant deployment scenario (See [Example from a Load Balanced and Highly Available Environment](#), which details an actual GIS Deployment). The implementation consists of multiple instances of GIS in a cluster, [as shown in the following figure](#).



GIS SOAP 7.6 HA deployment using a GIS cluster

With this deployment, if the current client-GIS connection fails:

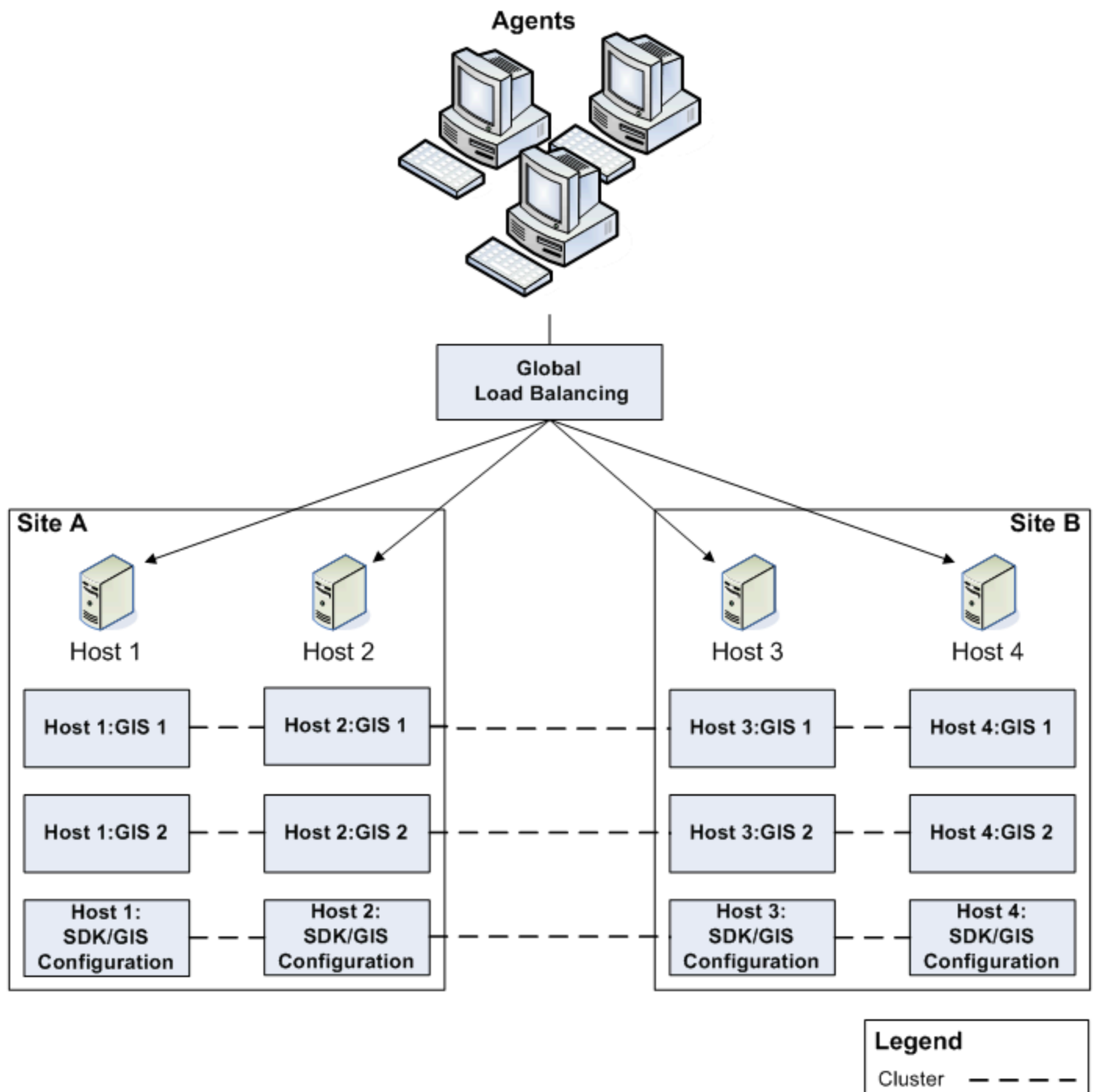
- The client application is automatically switched (by the dispatcher) to one of the other GIS nodes in the cluster.
- The session data is maintained by the cluster cache. Therefore, a database is not required.
- The agent session data is maintained. The agent does not need to log in again.

Important

For more information about this deployment, see [Deployment for High Availability with Load Balancing](#).

Example from a Load Balanced and Highly Available Environment

The following figure shows a multi-site, multi-host, multi-cluster, high availability environment that supports two-thousand (2000) agents.



Multi-site GIS cluster with HA

The environment includes:

- Agents
- Global Load Balancer (GLB)

- Multiple Sites (Site A and Site B)
- Multiple Host Machines (Hosts 1, Host 2, Host 3, and Host 4)
- Multiple GIS Instances (GIS 1, GIS 2, and SDK/GIS Configuration)
- Multiple GIS Clusters (GIS-cluster 1, GIS-cluster 2, and SDK/GIS Configuration cluster)

Agents

This environment supports a load of up to 2000 agents. The agent load is distributed over multiple sites and multiple hosts by GLB.

Global Load Balancer (GLB)

The GLB has a multi-dimensional configuration that distributes the agent load:

- Between two sites (four hosts).
- Across three GIS clusters.

To do this GLB is configured with three HTTP addresses (one per GIS cluster). When a request is received from an agent, a session is opened to one GIS cluster (only one!). Then, the sessions will be balanced between the four GIS instances in the cluster. For example, if one of the GIS instances becomes unavailable, the session is managed on the remaining instances in the cluster.

Important

GLB does not balance sessions across multiple GIS clusters. Once a session has been opened on a GIS cluster, it remains on that cluster.

Multiple Sites (Site A and Site B)

This sample environment is designed for multiple sites (Site A and Site B). This offers excellent availability. For example, if one site becomes unavailable, agents can connect to the second site and continue their work.

Multiple Host Machines (Host 1, Host 2, Host 3, and Host 4)

This design includes four host machines (two hosts per site). Each host has:

- Three GIS instances (GIS 1, GIS 2, and SDK/GIS Configuration)
- Two (2) GB for each GIS.

If a host or an application server on a host (Websphere) becomes unavailable, all agents connected to that host are redirected to another host by GLB.

Multiple GIS Instances (GIS 1, GIS 2, and SDK/GIS Configuration)

This sample environment includes three GIS instances per host:

- GIS 1—each instance of GIS 1 is configured to support up to 500 agents. It is the primary GIS on a host, and it is in cluster with the GIS 1s on the other host machines.
- GIS 2—each instance of GIS 2 is configured to support up to 500 agents. It is the backup instance on a host, and it is in cluster with the GIS 2s on the other host machines.
- SDK/GIS Configuration—supports ongoing configuration and any further SDK customization needed.

Multiple GIS Clusters (GIS 1 Cluster, GIS 2 Cluster, and SDK/GIS Configuration Cluster)

This sample environment includes three clusters:

- A cluster with four GIS 1s
- A cluster with four GIS 2s
- A cluster with four SDK/GIS Configurations

Typically, GLB directs agents to the GIS 1 Cluster. This cluster is replicated in the GIS 2 Cluster. However, GLB directs requests to the GIS 2 Cluster in the following circumstances:

- If the entire GIS 1 Cluster becomes unavailable (stops responding), agents connected to GIS 1 Cluster are redirected to GIS 2 Cluster.

If three-of-four host machines become unavailable (stop responding), agents are directed to either a GIS 1 or GIS 2 on the remaining host.

Licensing

GIS manages the license control for the Statistics, Configuration, and Interaction SDK Services, and manages the number of sessions allowed for each interface. GIS 7.6 requires License Manager 8.3, which is available on a separate CD.

When you purchase a license for a GIS SDK Service, you receive license keys for GIS, the appropriate SDK Service, and the Configuration Server Proxy interface (which was packaged separately in 6.x releases as CS Proxy).

Important

Both the Statistics and Configuration SDK Services require the Configuration Server Proxy license key to access configuration information. This includes the read-only configuration information for users who are purchasing only the Statistics SDK Service. If you are upgrading to GIS 7.6 from Genesys Interface Server 7.1 or earlier, you must migrate your existing license keys. For details, see the [Genesys Licensing](#)

Guide.

For more information on licensing, see:

- The [Genesys Licensing Guide](#) for a detailed discussion of Genesys product licensing and using License Manager.
- [Configuring the License Section of the Option Tab for GIS:SOAP](#), which contains instructions on how to configure license information.
- The *Developer's Guide* for each SDK Service. These explain how to configure your login message so that the client application is authorized to communicate with the correct SDK Service.

Backward Compatibility

GIS 7.6 servers are compatible with client applications written using Genesys 7.6, 7.5, 7.2 or 7.1 SDKs (Software Development Kits). However, features newly introduced in 7.6 will typically be available only if you also upgrade your client applications to incorporate new 7.6 SDK components.

Warning

The predecessor products, Genesys Interface Server 7.1 and Genesys .NET Server 7.1, are not compatible with client applications built using the Genesys 7.6 SDKs.

If you are upgrading from Genesys .NET Server 7.1, GIS 7.6 supports your existing client applications by exposing new versions of the same Genesys .NET services. You simply need to connect your client applications to GIS 7.6 instead of .NET Server 7.1.

If you are upgrading from Genesys Interface Server 7.1, GIS 7.6 supports your existing client applications by exposing new versions of all the same Genesys services.

With GIS release 7.6.6 and developments in .Net, while APIs have been preserved, proxies shipped for .NET have been migrated from 1.1 to 2.0. This requires customer applications to be recompiled against the new targeted framework (2.0 or higher).

Task and their Related Procedures

Tasks and their Related Procedures

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

Tasks and Related Procedures

Objective	Related Procedures and Actions
To install GIS as a SOAP stand-alone server or a SOAP web module on Solaris, HP-UX, Linux, AIX, or Tru64 UNIX platforms.	<p>You will have to import at least three application templates, one for Agent Interaction Service, one for Open Media Interaction Service, and one for GIS.</p> <p>You must create three application objects, one for Agent Interaction Service, one for Open Media Service, and one for GIS.</p> <p>For information regarding importing templates and creating application objects, see the Genesys Framework Deployment Guide.</p> <p>The following configuration procedures must be completed for each of your application objects:</p> <ul style="list-style-type: none">• Configure the General Tab.• Configure the Server Info Tab.• Configure the Start Info Tab.• Configure the Connections Tab.• Configure the Tenants Tab (for use with a multi-tenant environment). <p>For detailed information, refer to the Genesys Framework Deployment Guide.</p> <ul style="list-style-type: none">• Configure the Options Tab. <p>For detailed information on the options available for your application object, refer to:</p> <ul style="list-style-type: none">• Agent Interaction Service.• Open Media Interaction Service.• Genesys Integration Server. <p>Follow the steps in Installing GIS on a UNIX-Based Platform.</p>

Objective	Related Procedures and Actions
<p>To install GIS as a SOAP stand-alone server, or a SOAP web module on a Windows platform.</p>	<p>You will have to import at least three application templates, one for Agent Interaction Service, one for Open Media Interaction Service, and one for GIS.</p> <p>You must create three application objects, one for Agent Interaction Service, one for Open Media Service, and one for GIS.</p> <p>For information regarding importing templates and creating application objects, see the Genesys Framework Deployment Guide.</p> <p>The following configuration procedures must be completed for each of your application objects:</p> <ul style="list-style-type: none"> • Configure the General Tab. • Configure the Server Info Tab. • Configure the Start Info Tab. • Configure the Connections Tab. • Configure the Tenants Tab (for use with a multi-tenant environment). <p>For detailed information, refer to the Genesys Framework Deployment Guide.</p> <ul style="list-style-type: none"> • Configure the Options Tab. <p>For detailed information on the options available for your application object, refer to:</p> <ul style="list-style-type: none"> • Agent Interaction Service. • Open Media Interaction Service. • Genesys Integration Server. <p>Follow the steps in Installing GIS on a Windows Platform.</p>
<p>To run GIS in the BEA WebLogic application server.</p>	<ul style="list-style-type: none"> • Deploy GIS in the WebLogic application server. Deploying GIS in WebLogic. • Verify the successful deployment of GIS in WebLogic. Inspecting the WebLogic Directory Tree. • Confirm that the GIS server has started successfully in WebLogic. Checking the GIS Log Files.
<p>To run GIS in WebSphere application server.</p>	<ul style="list-style-type: none"> • Deploy GIS in the WebSphere. Deploying GIS in WebSphere. • Verify the successful deployment of GIS in WebSphere. Inspecting the WebSphere Directory Tree.

Objective	Related Procedures and Actions
	<ul style="list-style-type: none"> Confirm that the GIS server has started successfully in WebLogic. Checking the GIS Log Files.
To verify the successful installation of GIS:SOAP in Tomcat (stand-alone).	Inspecting the Stand-alone Directory Tree .
To uninstall GIS.	<ul style="list-style-type: none"> From a Windows platform for stand-alone server installations, Uninstalling GIS as a SOAP Stand-alone Server from a Windows Platform. From a UNIX-based platform for stand-alone server installations, Uninstalling GIS as a SOAP Stand-alone Server from a UNIX-Based Platform. From a Web application server, Uninstalling GIS from a WebSphere Application Server.
To deploy GIS with the High Availability option.	<ul style="list-style-type: none"> To create a standard node cluster without load balancing follow the procedures listed in Deployment for High Availability (No Load Balancing). To create a standard node cluster with load balancing follow the procedures listed in Deployment for High Availability with Load Balancing. To create a Primary/Backup GIS pair follow the procedures listed in Alternative Deployment for High Availability.
To customize your applications configuration.	<p>You can customize your application in various ways using the following procedures:</p> <ul style="list-style-type: none"> Windows Service Setting. LCA Configuration. Customize the WSDL Files. SDK Server Connections for GIS SOAP. Configuring a SOAP Port for a Master Configuration Server. Configuring a SOAP Port for a CS Proxy. Customizing the Notification Mechanism for GIS:SOAP Displaying the WSDL Files for GIS:SOAP.
To start/test GIS.	<p>Choosing Your Start Script Editing the Start Script. Starting and Testing GIS. For GIS:SOAP:</p> <ul style="list-style-type: none"> Specifying Licensing at Startup (GIS:SOAP only). Starting on UNIX-based Platforms (GIS:SOAP only)

Objective	Related Procedures and Actions
	<ul style="list-style-type: none"> • Verify the Status of GIS (GIS:SOAP Only). <p>For Windows Platforms:</p> <ul style="list-style-type: none"> • Starting GIS on Windows. <p>For Web Server Applications:</p> <ul style="list-style-type: none"> • Starting GIS on a Web Application Server. <p>Using Management Layer: Using Management Layer to Start and Stop GIS.</p>
To stop GIS.	<p>For Windows Platforms, follow Stopping GIS on Windows. For UNIX-based Platforms, follow Stopping GIS on UNIX-based Platforms. For Web Application Servers, follow Stopping on a Web Application Server. For Management Layer, follow Using Management Layer to Start and Stop GIS.</p>

Installing and Uninstalling

This chapter describes how to install, uninstall, and deploy the Genesys Integration Server (GIS) Application object.

GIS Installation and Deployment Options

The Genesys Integration Server can be installed and deployed on many different platforms and environments. Below you will find procedures for each of these options:

- To install GIS as a SOAP stand-alone server, or a SOAP web module on Solaris, HP-UX, Linux, or AIX platforms, see [Installing GIS on a UNIX-Based Platform](#).
- To install GIS as a SOAP stand-alone server, or a SOAP web module on Windows platforms: see [Installing GIS on a Windows Platform](#).
- To deploy GIS into an IBM WebSphere application server: see [Deploying GIS in WebSphere](#).
- To deploy GIS into the BEA WebLogic application server on the Solaris platform: see [Deploying GIS in WebLogic](#).

Installing GIS on a UNIX-Based Platform

Purpose

To install GIS as a SOAP stand-alone server, or a SOAP web module on Solaris, HP-UX, Linux, or AIX platforms.

Prerequisites

- Before selecting a GIS:SOAP as a Stand-alone Server installation:
 - Configure at least one GIS:SOAP application in the Configuration Layer; otherwise, the installer will abort when it cannot find an Application object.
- Before selecting a GIS:SOAP as a Web Module installation:
 - Define the JAVA_HOME environment variable to point to your Java SDK main directory. For instructions see, [Setting the JAVA_HOME Environment Variable](#).
 - Configure at least one GIS:SOAP application in the Configuration Layer; otherwise, the installer will abort when it cannot find an Application object. Refer to the Genesys [Framework 7.6 Deployment Guide](#) for information on how to create an application object.
 - Install and start your application server.

Important

Refer to the Genesys [Framework 7.6 Deployment Guide](#) for information on how to create an application object. Under WebLogic, you must deploy GIS as an exploded .war directory, not as a .war file. If you deploy GIS as a Web Module, you will not be able to use the Management Layer to start or stop GIS. However, you will be able to use the Management Layer to view GIS status.

- Before installing GIS on a Red Hat Enterprise Linux 6 64-bit operating system, you must first install the Red Hat compatibility packages.

Start

1. Open the appropriate folder for your platform on the installation CD. The folders are:
 - GIS/aix/
 - GIS/linux/
 - GIS/hp-ux/
 - GIS/solaris/
2. Copy all files in the folder to your home directory.
3. Locate and run the installation script, `install.sh`.
4. When prompted for the GIS component to install,
 - for GIS as a SOAP Stand-alone Server, select: 1) SOAP (SOAP Protocol).
 - for GIS as a SOAP Web Module, select: 1) SOAP (SOAP Protocol).
5. When prompted for the installation type,
 - for GIS as a SOAP Stand-alone Server, select: 1) Standalone server.
 - for GIS as a SOAP Web Module, select: 1) Web Module (Application Server deployment).
6. When prompted for the hostname, confirm or enter the GIS host information.
7. When prompted for your Configuration Server information, enter the Hostname, Network port, User name, and Password that you use to log into the Configuration Layer.
8. From the list of applications configured for this host, select a compatible application.

Important

If no applications, or incorrect applications, appear here, verify the `server-info-host` value that you configured in the GIS Configuration Layer application.

9. When prompted for the SOAP port:
 - for GIS as a SOAP Stand-alone Server, either accept the default or enter a different port number.
 - for GIS as a SOAP Web Module, the SOAP port prompt will not be displayed.

10. When prompted for the shutdown port:

- for GIS as a SOAP Stand-alone Server, either accept the default or enter a different port number.
- for GIS as a SOAP Web Module, the shutdown port prompt will not be displayed.

11. When prompted for the destination location for this installation:

- for GIS as a SOAP Stand-alone Server, specify the path.
- for GIS as a SOAP Web Module, the destination location prompt will not be displayed.
 - If you configured an installation path in your SOAP configuration object, you will be offered that path as a default.
 - If you are reinstalling to an existing directory, you will be offered a choice of backing up that directory's contents, selectively replacing them, or wiping them clean.

Important

GIS does not support installation paths that contain spaces.

12. When prompted for the target directory for the Web Application Archive:

- for GIS as a SOAP Stand-alone Server, the target directory for the Web Application Archive prompt will not be displayed.
- for GIS as a SOAP Web Module, enter the full path where you want the `gis.war` file placed.

13. Follow the script's remaining prompts to complete the installation.

End Next Steps

- To verify the installation of GIS, you can select from one of the following procedures:
 - [Inspecting the Stand-alone Directory Tree](#).
- To deploy your GIS:SOAP Web Module, choose one of the following procedures:
 - [Deploying GIS in WebSphere](#).
 - [Deploying GIS in WebLogic](#).

Installing GIS on a Windows Platform

Purpose

To install GIS as a SOAP Stand-alone Server or as a SOAP Web Module on Windows platforms.

Prerequisites

- Certain GIS services require a GIS:SOAP installation, see [GIS Overall Architecture diagram](#) and check whether your client application needs one of these services.

- Before selecting a SOAP Stand-alone Server:
 - Configure at least one GIS:SOAP application in the Configuration Layer; otherwise, the installer will abort when it cannot find an Application object.
- Before selecting a SOAP Web Module:
 - Define the JAVA_HOME environment variable to point to your Java SDK main directory. For instructions see, [Setting the JAVA_HOME Environment Variable](#).
 - Configure at least one GIS:SOAP application in the Configuration Layer. Refer to the Genesys Framework Deployment Guide for information on how to create an application object.

Important

If you deploy GIS as a Web Module, you will not be able to use the Management Layer to start or stop GIS. However, you will be able to use the Management Layer to view GIS status. This version of GIS supports WebLogic deployments only on Solaris.

Start

1. Open the GIS\windows\ folder on the installation CD.
2. Locate the setup.exe file and double-click it to start the Installation Wizard.
3. Click Next on the Welcome page to begin the installation.
4. If an instance of GIS already exists on this host, the Wizard presents a Maintenance Setup Type page. Select Install new instance of the application and click Next to install a new instance of GIS. If you choose to maintain an existing installation, you will be given the opportunity to uninstall that instance of GIS (see [Uninstalling GIS as a SOAP Stand-alone Server from a Windows Platform](#) for details).
5. On the Select installation type page, select:
 - for GIS as a SOAP Stand-alone Server—Deploy GIS.SOAP (SOAP protocol) as standalone server
 - for GIS as a SOAP Web Module—Deploy GIS.SOAP (SOAP protocol) as Web Module and click Next.
6. On the Connection Parameters to the Genesys Configuration Server page, specify the appropriate Host name, Port, User name, and Password. Click Next to continue.
7. On the Select Application page, choose your GIS Application. Click Next.
8. On the Choose Destination Location page, accept or override the default installation path. Click Next.

Important

GIS does not support installation paths that contain spaces.

9. On the Genesys Integration Server Parameters page, confirm or override:

- for GIS as a SOAP Stand-alone Server—the default Server Port and Shutdown Port for GIS:SOAP and click Next.

10. Click **Install** to begin the installation (see [Windows Service Setting](#).)

11. Click **Finish** once the installation is complete, to exit the Wizard.

End Next Steps

- To verify the installation of GIS, you can select from one of the following procedures:
 - [Inspecting the Stand-alone Directory Tree](#).
- To deploy your GIS:SOAP Web Module, choose one of the following procedures:
 - [Deploying GIS in WebSphere](#).
 - [Deploying GIS in WebLogic](#).

Deploying GIS in WebLogic

Purpose

To deploy GIS in the BEA WebLogic application server on Solaris.

Prerequisites

- You must have GIS installed on your Solaris platform. Use [Installing GIS on a UNIX-Based Platform](#), as a guide.

Important

In this release, deployment into WebLogic is supported on the Solaris platform only.

If you are deploying the GIS:SOAP connector in WebLogic 8.1, you *must* do so using an exploded .war directory, as instructed below.
Deploying a

.war (web archive) file for this combination will *not* produce correct results.

Start

1. Locate the `gis.war` file created by the GIS installer.

For UNIX-based installations, this file is located in the directory specified in See When prompted for the target directory for the Web Application Archive: of [Installing GIS on a UNIX-Based Platform](#).

2. Unpack (“explode”) the `gis.war` file’s contents into a directory or subdirectory named `gis_76`.

Important

If your unpacking utility does not recognize the .war file, try changing that file's extension to a .zip or .jar.

- Start the WebLogic Server Administration Console for the domain in which you will be working.
- In the WebLogic Server Console's navigation tree (left pane), expand Deployments, right-click Web Application Modules, and select Deploy a new Web Application Module. This initiates the Deployment Assistant.
- Use the Deployment Assistant to select the gis_76 directory/subdirectory into which you exploded gis.war's contents in Step 2 above. The WebLogic Server will deploy all components that it finds in the selected directory and its subdirectories.
- Click Target Application.
- If your domain contains more than one server or (load-balancing) cluster, and you are prompted to Select targets for this Web Application Module: select the server or cluster onto which you want to deploy GIS, then click Continue. If you have only one server in your domain, ignore this step.
- Enter gis in the Name field as the name for the Web Application.

Important

If you have more than one server or cluster in your domain, click the appropriate radio button to select whether you want to copy the file to each server. (Remember to change the application name for each copy in that copy's own ../gis_76/conf/gis.properties file.)

- Click Deploy. The WebLogic Server Console displays the Deploy panel, which lists deployment status and deployment activities for the Web Application. After a few seconds, the status will change to Success.
- Point your browser to the following URL and verify that the GIS connector is starting:

`http://<your_deploymentserver_host>:<your_deploymentserver_port>/gis`

End Next Steps

- If you want to confirm that GIS has started successfully [Checking the GIS Log Files](#).
- To confirm that the installation of GIS in WebLogic was correct and complete, [Inspecting the WebLogic Directory Tree](#).
- To customize your GIS environment, [Customizing the GIS Environment](#).
- To add High Availability and Load Balancing to your GIS environment, [High-Availability Deployments of GIS](#).
- To start and test GIS, [Starting and Testing GIS](#).

Deploying GIS in WebSphere

Purpose

To deploy GIS in WebSphere regardless platform type.

Prerequisites

- You must have and GIS installed. Use the procedures below to install GIS according to your platform requirements:
 - [Installing GIS on a UNIX-Based Platform.](#)
 - [Installing GIS on a Windows Platform.](#)

Start

1. Open the WebSphere Administration Console, as follows:
 - For WebSphere 6.0.x, the default URL is: `http://<WebSphere_host>:9061/ibm/console/`
 - For WebSphere 7.0.x, the default URL is: `http://<WebSphere_host>:9043/ibm/console/`

In the above examples, <WebSphere_host> represents the host where the WebSphere application server is running.

Important

If you need to modify the startup parameters, such as the CS host name or the CS port number, locate and edit the `gis.properties` file. Within your WebSphere directory structure, this file is located in the following subdirectory:

```
.../AppServer/installedApps/<node_name>/<application_name>/gis.war/conf/
```

2. In the Administration Console's navigation tree, click Applications > Install New Application. This displays the first of two Preparing for application install pages.
3. On the first Preparing for application install page, specify the full path name of the GIS application source file (the .war file). The GIS .war file can be either on the client machine (the machine that runs the web browser) or on the server machine (the machine to which the client is connected). If you specify a .war file on the client machine, the Administration Console uploads the .war file to the machine on which the Console is running, and proceeds with application installation.
4. On the Provide options to perform the installation page, click Next.
5. On the Map virtual hosts for web modules page, click Next.
6. On the Map modules to application servers page, select a target server or cluster for the GIS module from the Clusters and Servers list. Select the check box beside GIS Module.
7. On the Summary page, verify the cell, node, and server onto which the GIS application module will install.
8. Beside the Cell/Node/Server option, click Click here and verify the settings.
9. Click Finish.
10. On the Administration Console's taskbar, click Save to save the changes to your configuration. This registers the application with the administrative configuration, and copies application files to the target directory.

End Next Steps

- If you are deploying GIS in WebSphere 5.0, you must complete the following upgrade procedure. [Replacing the JavaMail Archive \(for WebSphere 5.0\)](#).
- If you are not deploying in WebSphere 5.0 then you must verify your installing by following the [Inspecting the WebSphere Directory Tree](#) procedure.
- To customize your GIS environment, [Customizing the GIS Environment](#).
- To add High Availability and Load Balancing to your GIS environment, [High-Availability Deployments of GIS](#).
- To start and test GIS, [Starting and Testing GIS](#).

Inspecting the Stand-alone Directory Tree

Purpose

To confirm that the installation of GIS as a SOAP Stand-alone Server was correct and complete.

Prerequisites

- You must have installed GIS as a SOAP Stand-alone Server on either your UNIX-based or Windows platform. You can do so by following the steps in the provided:
 - For UNIX-based platforms: [Installing GIS on a UNIX-Based Platform](#).
 - For Windows platforms: [Installing GIS on a Windows Platform](#).

Start

1. Locate and inspect the directory structure created during the Installation procedure. The directory structure is located under the top-level installation directory (for example, under /gcti/IntegrationSDK7.5/GIS/server/SOAP):
 - bin/ contains all scripts used to start or stop the server, to set up environment variables, and to start or stop GIS as a Windows service.
 - conf/ contains all files used to configure Tomcat.
 - common/ contains various jar files that Tomcat uses.
 - logs/ contains log files that GIS and the Management Layer create.
 - server/ contains various jar files that Tomcat uses.
 - webapps/ contains the GIS web application directories and files:
 - gis/ the GIS application directory contains the following subdirectories:
 - conf/ contains all the configuration files that GIS uses.
 - html/ contains HTML pages for diagnostic reporting.
 - images/ contains the logo and various images used on the GIS home page.
 - licenses/ contains the licenses of the third-party toolkit that GIS uses.

- WEB-INF/ contains:
 - classes/ contains precompiled .jsp files.
 - lib/ contains binary files that GIS uses.
 - soa/ail/ contains .jar and deployment-descriptor files for the Agent Interaction services.
 - soa/openmedia/ contains .jar and deployment-descriptor files for the Open Media Interaction services.
 - server-config.wsdd configures the various services in GIS.
 - soa.properties contains properties related to the service-oriented architecture, through which GIS exposes services.
 - web.xml provides configuration and deployment information for the web application.

Warning

The directory tree listed above is required for the scripts and configuration files to work properly. Do not alter it.

End Next Steps

- To customize your GIS environment, [Customizing the GIS Environment](#).
- To add High Availability and Load Balancing to your GIS environment, [High-Availability Deployments of GIS](#).
- To start and test GIS, [Starting and Testing GIS](#).

Inspecting the WebSphere Directory Tree

Purpose

To confirm that the installation of GIS in WebSphere was correct and complete.

Prerequisites

- You must have deployed GIS in a WebSphere using the steps provided in [Deploying GIS in WebSphere](#).

Start

1. Inspect the directory tree. Installing GIS creates the following directory structure located under the top-level installation directory (which might be, for example, <WebSphere_home>/AppServer/):
installedApps/<node_name>/gis.war contains the GIS web application directories and files:
 - gis.war/ the GIS application directory contains the following subdirectories:

- `conf/` contains all the configuration files that GIS uses.
- `html/` contains HTML pages for diagnostic reporting.
- `images/` contains the logo and various images used on the GIS home page.
- `licenses/` contains the licenses of the third-party toolkit that GIS uses.
- `META-INF/` contains the GIS archive's manifest file.
- `WEB-INF/` contains:
 - `classes/` contains precompiled `.jsp` files.
 - `lib/` contains binary files that GIS uses.
 - `soa/ail/` contains `.jar` and deployment-descriptor files for the Agent Interaction services.
 - `soa/openmedia/` contains `.jar` and deployment-descriptor files for the Open Media Interaction services.
 - `ibm-web-bnd.xmi` contains WebSphere binding information.
 - `ibm-web-ext.xmi` contains WebSphere extensions information.
 - `server-config.wsdd` configures the various services in GIS.
 - `soa.properties` contains properties related to the service-oriented architecture, through which GIS exposes services.
 - `web.xml` provides configuration and deployment information for the web application.

Warning

The directory tree listed above is required for the scripts and configuration files to work properly. Do not alter it.

End Next Steps

- To customize your GIS environment, [Customizing the GIS Environment](#).
- To add High Availability and Load Balancing to your GIS environment, [High-Availability Deployments of GIS](#).
- To start and test GIS, [Starting and Testing GIS](#).

Inspecting the WebLogic Directory Tree

Purpose

To confirm that the installation of GIS in WebLogic was correct and complete.

Prerequisites

- You must have deployed GIS in a WebLogic using the steps provided in [Deploying GIS in WebLogic](#).

Start

1. Inspect the directory tree. Installing GIS creates the following directory structure located under your WebLogic home directory:
 - `conf/` contains all the configuration files that GIS uses.
 - `html/` contains HTML pages for diagnostic reporting.
 - `images/` contains the logo and various images used on the GIS home page.
 - `licenses/` contains the licenses of the third-party toolkit that GIS uses.
 - `WEB-INF/` contains:
 - `classes/` contains precompiled `.jsp` files.
 - `lib/` contains binary files that GIS uses.
 - `soa/ail/` contains `.jar` and deployment-descriptor files for the Agent Interaction services.
 - `soa/openmedia/` contains `.jar` and deployment-descriptor files for the Open Media Interaction services.
 - `server-config.wsdd` configures the various services in GIS.
 - `soa.properties` contains properties related to the service-oriented architecture, through which GIS exposes services.
 - `web.xml` provides configuration and deployment information for the web application.
 - `weblogic.xml` provides configuration and deployment information for the web application.

The directory tree listed above is required for the scripts and configuration files to work properly. Do not alter it.

End Next Steps

- If you want to confirm that GIS has started successfully, see [Checking the GIS Log Files](#).
- To customize your GIS environment, see [Customizing the GIS Environment](#).
- To add High Availability and Load Balancing to your GIS environment, see [High-Availability Deployments of GIS](#).
- To start and test GIS, see [Starting and Testing GIS](#).

Checking the GIS Log Files

Purpose

To check the GIS log files to confirm that your GIS server has started successfully.

Prerequisites

- You must have deployed GIS in a WebLogic using the steps provided in [Deploying GIS in WebLogic](#).

Start

1. In the WebLogic Server Console's navigation tree, select the server that you created in See Enter gis in the Name field as the name for the Web Application. on See Enter gis in the Name field as the name for the Web Application..
2. Select the Control tab.
3. Select Remote Start Output .
4. Select View Server Output .

End

Next Steps

- To customize your GIS environment, [Customizing the GIS Environment](#).
- To add High Availability and Load Balancing to your GIS environment, [High-Availability Deployments of GIS](#).
- To start and test GIS, [Starting and Testing GIS](#).

Uninstalling GIS as a SOAP Stand-alone Server from a Windows Platform

Purpose

To uninstalling GIS as SOAP Stand-alone Server from a Windows Platform.

Prerequisites

- You must have already installed GIS as a SOAP Stand-alone Server using one of the following procedure, [Installing GIS on a Windows Platform](#).

Start

1. Stop GIS. (See [Stopping GIS](#) for details.)
2. From the Windows Control Panel, launch the Add/Remove Programs applet, select Genesys Integration Server, and select Remove .
3. Manually remove any files that were not created by the installation script, including: log files, the license file, and any other files you may have installed to support GIS and the Statistics SDK, Configuration SDK, or Interaction SDK Services.

End

Uninstalling GIS as a SOAP Stand-alone Server from a UNIX-

Based Platform

Purpose

To uninstall GIS as a SOAP Stand-alone Server from a UNIX-based Platform.

Prerequisites

- You must have already installed GIS as a SOAP Stand-alone Server using one of the following procedure:
[Installing GIS on a UNIX-Based Platform](#).

Start

1. Stop GIS. (See [Stopping GIS](#) for details.)
2. Open the GIS home directory and manually remove all files.

End

Uninstalling GIS from a WebSphere Application Server

Purpose

To uninstall GIS from a WebSphere application server.

Prerequisites

- You must have deployed GIS in WebSphere using [Deploying GIS in WebSphere](#).

Start

1. If you choose to use the provided Websphere Uninstall Script instead of using the recommended WebSphere Administration Console, navigate to the WebSphere server /bin subdirectory.
2. At the command line, enter one of the following commands, depending on your operating system. Here, <ApplicationName> represents the name that you originally used to deploy GIS into the web application server:

Windows: `wsadmin.bat -c "$AdminApp uninstall <ApplicationName>"`
UNIX: `wsadmin.sh -c "\$AdminApp uninstall <ApplicationName>"`

End

Next Steps

- This uninstall procedure does *not* remove any files that were not created by the original installation script—such as the log files, license file, and any other files you may have installed to support the Genesys Statistics SDK, Configuration SDK, and/or Interaction SDK products. You must remove these files manually.

Uninstalling GIS from a WebLogic Application Server

Purpose

To uninstall GIS from a WebLogic application server.

Prerequisites

- You must have deployed GIS in WebLogic using [Deploying GIS in WebLogic](#).

Start

1. In the navigation tree of the WebLogic Server Console (left pane), expand Deployments.
2. Right-click on the Web Application Modules.
3. Select the existing GIS Web Application Module.
4. On the right pane, select the Deploy tab and stop the associated server.
5. Once stopped, right click on the GIS Web Application Module on the left pane and select “Delete ...”.
6. Confirm the deletion of GIS by clicking the Yes button on the right pane.

End

Next Steps

- This uninstall procedure does *not* remove any files that were not created by the original installation script—such as the log files, license file, and any other files you may have installed to support the Genesys Statistics SDK, Configuration SDK, and/or Interaction SDK products. You must remove these files manually.

Setting the JAVA_HOME Environment Variable

Purpose

To set the JAVA_HOME environment variable for your specific operating system. This will allow your operating system to reference the absolute path to the JRE or Java SDK main directory.

Important

For UNIX-based platforms, you must also set the value of the JAVA_HOME environment variable to the absolute path to the JRE or Java SDK main directory. The actual steps to set an environment variable can vary depending on the platform and shell being used.

Start

1. On Windows 2000 or Windows 2003, open your Windows Control Panel.

2. Double-click System to open the System Properties dialog box.
3. Select the Advanced tab.
4. Click the Environment Variables button.
5. Locate the JAVA_HOME entry in the System Variables pane.

If the JAVA_HOME entry does not exist, then click NEW .
Otherwise, select the JAVA_HOME entry and click EDIT .

6. Enter the variable name and path to your Java installation in the available text boxes.

If you have deployed GIS as a Web Module, point to your Java SDK main directory.

7. Click OK to save the variable settings.
8. Click OK to close the Environment Variables dialog box.
9. Click OK to close the System dialog box.

End Next Steps

- Create and configure an instance of GIS. See the [Framework 8.1 Deployment Guide](#) for details.

Customizing the GIS Environment

This chapter describes how to integrate GIS with the Genesys Management Layer, how to customize the statistical notification mechanism, how to optimize GIS functionality, and other configuration considerations.

Windows Service Setting

If you deploy GIS as a SOAP, GIS installs as a Windows service.

Warning

Do not change this setting; changes are not supported.

LCA Configuration

To integrate GIS with the Genesys Management Layer's Solution Control Server (SCS), you must install Local Control Agent (LCA), version 7.0.100.05 or higher, on the host that supports GIS. For LCA installation and configuration instructions, see the [Framework 8.1 Deployment Guide](#).

Important

If you have installed GIS within the WebSphere or WebLogic application server, you cannot use the Management Layer to start or stop GIS. Doing so would crash the application server. However, you can use LCA to monitor GIS status (that is, Primary versus Backup). For a stand-alone GIS installation, LCA is required for Management Layer integration.

Tuning the Heap Size Values

To ensure maximum performance of your applications, you must set your heap size values to minimize the time that your virtual machine spends doing garbage collection while maximizing the number of clients that your server can handle at a given time. This can be achieved by adjusting the size of the memory allocation pool using the following options:

- `xms` —This value represents the minimum size of the pool

- `xmx` —This value represents the maximum size of the pool

Tuning Heap Size Values

Purpose

To set the minimum and maximum sizes of the memory allocation pool for GIS:SOAP deployments.

Start

1. Open the `setIni.bat` file for Windows or the `gis_soap.sh` file for Unix. These files are located under the `bin` directory for GIS:SOAP.
2. Search for and set the `xmx/xms` parameters according to your environment.

End

SDK Server Connections for GIS SOAP

Connections and Application objects for the Configuration SDK Service depend on what version and configuration you are using. For details, see [Configuration SDK Server Connections](#).

Configuration SDK Server Connections

Configuration Server Version	Application Objects and Connections
7.6, 7.5 or 7.2	<ul style="list-style-type: none">• If you are using a master Configuration Server, GIS automatically detects and connects to the SOAP interface if it exists. If no SOAP port is detected then you should configure a <code>CS_Proxy</code> application and add it to the Connections tab.• If you configured a CS Proxy application, click Add to specify that application on the GIS Connections tab.
7.1 or 7.0	<ul style="list-style-type: none">• If it does not already exist, create an Application object using the Genesys Interface Server template.• If you are using a master Configuration Server, GIS automatically detects and connects to the SOAP interface if it exists. If no SOAP port is detected then you should configure a <code>CS_Proxy</code> application and add it to the Connections tab.• If you configured a CS Proxy application, click

Configuration Server Version	Application Objects and Connections
	Add to specify that application on the GIS Connections tab.
6.5	<ul style="list-style-type: none">• If it does not already exist, create an Application object using the Genesys Interface Server template.• Click Add and specify the CS Proxy 6.5 application on the GIS Connections tab. You must use a CS_Proxy application with pre-7.x versions

Configure the Configuration Server SOAP Port

To access Configuration Server using the SOAP interface of the Configuration SDK Service, you need to configure a SOAP port. You must do this even if you have purchased only the Statistics SDK Service, but need the Configuration SDK Service for read-only access to configuration information. The procedure to configure the SOAP port depends on the version and settings of your Configuration Server.

- Configuration Server 7.x incorporates the SOAP interface that was packaged in release 6.5 as CS Proxy. If you are using Configuration Server 7.x, you can choose to configure any instance of Configuration Server either as a master Configuration Server or as a CS Proxy.
- If you are using Configuration Server 6.5, the SOAP port is available only from the CS Proxy application. In this case you must create a CS Proxy application, and use Configuration Manager to connect to the CS Proxy from the GIS Application object's Connections tab. Refer to the Genesys Framework Deployment Guide for information on how to create and configure an application object.

Important

See your [Framework documentation](#) for additional details about configuring Configuration Server options.

Configuring a SOAP Port for a Master Configuration Server

Purpose

To configure a SOAP interface port for a Configuration Server 7.x that has been configured as a master Configuration Server.

Start

1. Locate the configuration file. It resides in the same folder specified in the Start Info > Working Directory field of the Configuration Server's Application object. The file is named `confserv.conf` on UNIX platforms and `confserv.cfg` on Windows platforms.
2. Open the configuration file of the appropriate Configuration Server.
3. Locate the soap section in the configuration file. Initially, this section contains a single option that does not have its value set: `port = [ToBeChanged: soap_port]`
4. Update the soap section by adding or modifying parameters in the soap section of the configuration file, as described below:
 - `port`: The value for the listening port for the SOAP interface.
 - `debug`: If set to Yes, Configuration Server prints all the data being exchanged through the SOAP port to the log. The default value is No.
 - `client_lifespan`: Sets the time (in seconds) for Configuration Server to keep information about a closed SOAP connection; that is, the value of a cookie. The connection can be restored within the specified time interval to create continuous HTTP sessions. The default value is 600 seconds.
5. Save and close the configuration file.

End Next Steps

- Customize your GIS application further by:
 - [Windows Service Setting](#).
 - [LCA Configuration](#).
 - [SDK Server Connections for GIS SOAP](#).
 - [Configure the Configuration Server SOAP Port](#).
 - [Customize the Notification Mechanism](#).
 - [Customize the WSDL Files](#).

Configuring a SOAP Port for a CS Proxy

Purpose

To configure a SOAP port for a Configuration Server 7.x that has been configured as a CS Proxy.

Start

1. Navigate to your CS Proxy application in Configuration Manager and double-click it.
2. Click the Options tab in the Properties window that is displayed.
3. Double-click the soap section.
4. Right-click below the existing soap options, then select New from the shortcut menu.
5. In the Edit Option window, create a new option with the following parameters:
 - Option Name: `port`
Option Value: The value for the listening port for the SOAP interface.
6. Click OK.

Important

Since the `Configuration_Server_Proxy.apd` template does not contain a port option, you must add that option as described above.

End Next Steps

- Customize your GIS application further by:
 - [Windows Service Setting](#).
 - [LCA Configuration](#).
 - [SDK Server Connections for GIS SOAP](#).
 - [Configure the Configuration Server SOAP Port](#).
 - [Customize the Notification Mechanism](#).
 - [Customize the WSDL Files](#).

Customize the Notification Mechanism

You can edit the GIS:SOAP configuration file `modules.conf` to customize the statistical notification mechanism to meet your needs.

Customizing the Notification Mechanism for GIS SOAP

Purpose

To edit the GIS:SOAP configuration file `modules.conf` to customize the statistical notification mechanism to meet your needs.

Prerequisites

- Genesys suggest that you refer to “Notification Modes” in the [Statistics SDK 7.6 Web Services API Reference](#)’s “About the Statistics SDK Service” chapter for a description of the statistical notification mechanism.
- Review the list of `modules.conf` options that you can change to customize the notification mechanism found at [Statistical Notification Configuration Options](#).

Start

1. Navigate to the GIS:SOAP configuration file `modules.conf`, whose default location depends on your installation type:
 - Stand-alone: `<GIS_home>/webapps/gis/conf/modules.conf`
 - WebSphere: `<GIS_home>/gis.war/conf/modules.conf`

2. Navigate to the notification element of this file.
3. Modify the notification element to include two custom tags: debug and maxAttempts. For example:

```
<module name="notification"
class="com.genesyslab.gis.modules.notification.NotificationModule">
  <maxThreads>50</maxThreads>
  <minThreads>10</minThreads>
  <priority>7</priority>
  <http_version>HTTP 1.1</http_version>
  <debug>true</debug>
  <maxAttempts>5</maxAttempts>
</module>
```

End Next Steps

- Review the list of modules.conf options that you can change to customize the notification mechanism found at [Statistical Notification Configuration Options](#).
- Customize your GIS application further by:
 - [Windows Service Setting](#).
 - [LCA Configuration](#).
 - [SDK Server Connections for GIS SOAP](#).
 - [Configure the Configuration Server SOAP Port](#).
 - [Customize the Notification Mechanism](#).
 - [Customize the WSDL Files](#).

Statistical Notification Configuration Options

Option Name	Value	Description
maxThreads	Default Value: 50 Valid Values: integer > 0	Maximum number of threads allowed in the notification queue.
minThreads	Default Value: 10 Valid Values: integer > 0	Minimum number of threads created at the beginning in the notification queue.
priority	Default Value: 7 Valid Values: integer > 0	Sets the priority for each thread in the queue.
http_version	Default Value: "HTTP 1.1" Valid Values: "HTTP 1.1" or "HTTP 1.0"	Hypertext Transfer Protocol version used for notifications.

Option Name	Value	Description
debug	Default Value: true Valid Values: true or false	Boolean. If true, GIS displays information about notification. (This tag is not present in the default modules.conf file—add it as needed.)
maxAttempts	Default Value: 5 Valid Values: integer > 0	Maximum number of times that GIS will try to send notifications to a web server that does not respond. (This tag is not present in the default modules.conf file—add it as needed.)

The HTTP Connection Element

The HTTP Connector element represents a Connector component that supports the HTTP/1.1 protocol. It enables Catalina to function as a stand-alone web server, in addition to its ability to execute servlets and JSP pages. A particular instance of this component listens for connections on a specific TCP port number on the server.

At server startup time, this Connector will create a number of request processing threads (based on the value configured for the minSpareThreads attribute). Each incoming request requires a thread for the duration of that request. If more simultaneous requests are received than can be handled by the currently available request processing threads, additional threads will be created up to the configured maximum (the value of the maxThreads attribute). If still more simultaneous requests are received, they are stacked up inside the server socket created by the Connector, up to the configured maximum (the value of the acceptCount attribute). Any further simultaneous requests will receive connection refused errors, until resources are available to process them.

Setting the HTTP Connector Thread Number for GIS SOAP

Purpose

To set the maximum number of simultaneous requests that can be handled by one GIS:SOAP server.

Start

1. Open the conf/server.xml file.
2. Update the maxThreads attribute defined in the Connector item:

```
<Connector port="<GIS_PORT>" maxHttpHeaderSize="8192"
maxThreads="500" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" redirectPort="8443" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true"/>
```

End

Customize the WSDL Files

You can customize your service to display the WSDL files using the following procedure.

Displaying the WSDL Files for GIS SOAP

Purpose

To generate the WSDL files for each SDK Service.

Start

1. Open the GIS web page.
2. Click the `wSDL` link associated with the appropriate service. The corresponding WSDL file appears in the right-hand pane of the browser.

End

Or you can:

Start

1. Open any browser.
2. Enter the appropriate URL:
 - **Session**—`http://<GIS_HOST>:<GIS_PORT>/gis/services/SessionService?wsdl`
 - **Statistics**—`http://<GIS_HOST>:<GIS_PORT>/gis/services/StatService?wsdl`
 - **Configuration**—`http://<GIS_HOST>:<GIS_PORT>/gis/services/CSProxyService?wsdl`
 - **Agent Interaction**—one of:
 - `http://<GIS_HOST>:<GIS_PORT>/gis/services/AIL_<service_name>?wsdl` [for GIS 7.1 or later clients]
 - `http://<GIS_HOST>:<GIS_PORT>/gis/services/interaction/<service_name>?wsdl` [for backward compatibility with GIS 7.0 clients]
 - **Open Media Interaction**—`http://<GIS_HOST>:<GIS_PORT>/gis/services/OPENMEDIA_<service_name>?wsdl`

End

Next Steps

- Customize your GIS application further by:
 - [Windows Service Setting](#).
 - [LCA Configuration](#).
 - [SDK Server Connections for GIS SOAP](#).
 - [Configure the Configuration Server SOAP Port](#).
 - [Customize the Notification Mechanism](#).

- [Customize the WSDL Files.](#)

GZIP Compression

GIS:SOAP can consume a lot of bandwidth due to its nature (XML messages on HTTP). Large messages sent from the web container (Tomcat, or others) can be zipped and un-zipped by using GZIP.

Enabling GZIP Compression for .NET

Purpose

To enable GZIP compression for .NET, enabling you to zip/unzip large messages sent from the web container.

Prerequisites

- Requires .NET 2.0 or higher.

Start

1. Locate the `ail-configuration.xml` file.
2. Add the following option:

```
<option name="EnableDecompression" type="bool" value="true"/>
```

3. If `EnableHttpCommonsSender` is valid on Tomcat, add the following to the `server.xml` file:

```
<Connector port="8080"
protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443"
compression="on"
compressionMinSize="1"
noCompressionUserAgents="gozilla, traviata"
compressableMimeType="text/html,text/xml"/>
```

End

Next Steps

- Customize your GIS application further by:
 - [Windows Service Setting.](#)
 - [LCA Configuration.](#)
 - [SDK Server Connections for GIS SOAP.](#)
 - [Configure the Configuration Server SOAP Port.](#)
 - [Customize the Notification Mechanism.](#)
 - [Customize the WSDL Files.](#)

Enabling GZIP Compression for Java

Purpose

To enable GZIP compression for Java, enabling you to zip/unzip large messages sent from the web container.

Start

1. Locate the proxy-configuration file.
2. Add the following option:

```
AcceptGZIP = true
```

3. If EnableHttpCommonsSender is valid on Tomcat, add the following to the server.xml file:

```
<Connector port="8080"
protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443"
compression="on"
compressionMinSize="1"
noCompressionUserAgents="gozilla, traviata"
compressableMimeType="text/html,text/xml"/>
```

End

Next Steps

- Customize your GIS application further by:
 - [Windows Service Setting](#).
 - [LCA Configuration](#).
 - [SDK Server Connections for GIS SOAP](#).
 - [Configure the Configuration Server SOAP Port](#).
 - [Customize the Notification Mechanism](#).
 - [Customize the WSDL Files](#).

Enabling GZIP Compression using Web Server Features

Purpose

To enable GZIP compression using the Web Server features when Tomcat is used behind the Web Server.

Important

If Tomcat is used behind a Web Server, the AJP connector must be used. Unfortunately, the AJP connector has no compression support. Instead, the compression features of the Web Server can be used.

Start

1. For IIS, Apache, and so on, refer to the respective documentation to enable response compression by using gzip.

Important

If the compression rate is between 1.5 and 8, the compression will increase CPU load by 5%. Refer to the following example as a guide:

```
getInteraction DTO w attachment 1MBytes .zip file, w/o compression 1379913 bytes,  
w compression 995729 bytes getInteraction DTO w attachment 1MBytes .ppt file,  
w/o compression 1355049 bytes,  
w compression 791880 bytes getInteraction DTO w/o attachment,  
  
w/o compression 8371bytes,  
w compression 1167 bytes
```

End Next Steps

- Customize your GIS application further by:
 - [Windows Service Setting](#).
 - [LCA Configuration](#).
 - [SDK Server Connections for GIS SOAP](#).
 - [Configure the Configuration Server SOAP Port](#).
 - [Customize the Notification Mechanism](#).
 - [Customize the WSDL Files](#).

High-Availability Deployments of GIS

This chapter outlines how to deploy GIS in high-availability (HA) mode, and describes the load balancing capabilities of those deployments.

Overview

In a High Availability deployment of GIS 7.6, when an instance of GIS fails, all of the session data is adopted by another instance of GIS. This means that the sessions, agent logins, event subscribers, and statistics associated with that failed GIS are then used by another instance. In this way client requests and events are restored seamlessly.

Important

In HA, during the GIS switchover, when working with a SIP Server, the DN is logged out if the SIP Server option,

logout-on-disconnect is set to true

.

GIS HA may be setup using either:

- The Primary/Backup pair model. This model includes two instances of GIS (one configured as the primary server and one configured as the backup server) and a database where the information is stored. Either one of the two GIS instances could act as the failover instance for a failed GIS (See [Alternative Deployment for High Availability](#)).
- The Cluster model. This model includes multiple instances of GIS all sharing context information in an embedded/configurable distributed in-memory cache. Any one of the GIS instances could act as the failover instance for a failed GIS (See [Deployment for High Availability \(No Load Balancing\)](#), and [Deployment for High Availability with Load Balancing](#)).

Important

The failover scenario must occur with homogeneous sets of connectors—that is, all connectors must be configured and installed as the same type of connectors.

The essence of this cluster model is that any instance (or node) in a cluster can take over the role for any failed GIS. This also means that there is no backup mode for GIS; it does not switch to primary since it has neither backup nor primary state to begin with. GIS 7.6 can always accept client

connections—something that previous versions of GIS did not offer.

Important

Machines hosting GIS nodes must have their clocks synchronized.

Genesys recommends that you use the cluster configuration where all instances of GIS are active all of the time, enabling each instance to be a potential backup of the other(s). Clusters enable scalability in high-availability load-balanced environments.

Deployment for High Availability (No Load Balancing)

GIS-HA mode may be deployed without or without load balancing functionality. This section describes GIS-HA deployment without load balancing. For details about GIS-HA deployment with load balancing (See [Deployment for High Availability with Load Balancing](#)).

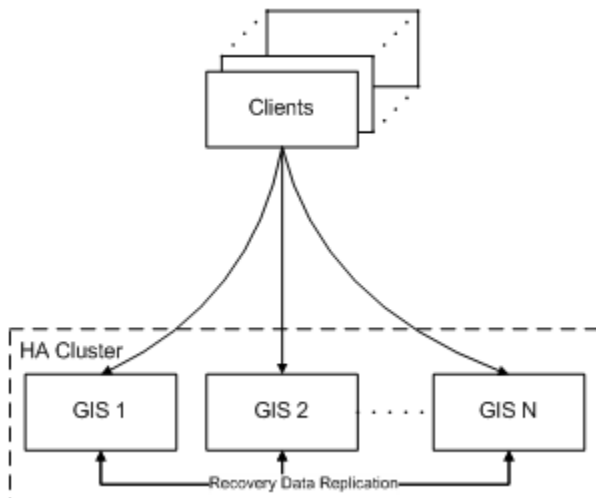
A standard HA deployment uses a cluster of GIS nodes, where each node is a GIS instance that has been fully configured and installed as described in:

- [Installing and Uninstalling GIS](#).

There are 3 main procedures that must be followed to deploy GIS in HA mode:

- [Configuring the GIS Application Objects](#)
- [Configuring the HA Cluster for Manual Peer Discovery Support](#)
- [Configuring the Client Application](#)

[High Availability Cluster](#) describes a GIS HA cluster with connections to clients.



High Availability Cluster

Configuring the GIS Application Objects

Purpose

To configure your GIS application objects for High Availability deployment.

Prerequisites

- You must have completely configured and installed your GIS nodes using the procedures found in [Installing and Uninstalling GIS](#).

Start

For each node, use Configuration Manager to configure one GIS application using the following steps:

1. Open the GIS Application object in Configuration Manager.
2. Select the Server Info tab.
3. Ensure that the Redundancy Type is set to Warm or Hot Standby , and that the Backup Server is set to None .

Other considerations you should keep in mind include:

- To use Agent Interaction and Open Media services, each GIS must be connected to its own Agent Interaction and/or Open Media application.
- Use different ports for each application on a host.

End

Next Steps

- Configure the HA cluster by following the steps in [Configuring the HA Cluster for Manual Peer Discovery Support](#).

Or:

- Configure the HA cluster by following the steps in [Configuring the HA Cluster for Automatic Peer Discovery Support](#).

Important

When choosing either manual or automatic peer discovery support, you may want to consider the following points.

Automatic peer discovery is:

- more easily extendable, since the ehcache.xml file does not require any changes to add more peers (more GIS instances) to the cluster.
- more traffic-efficient, since peer discovery messages are multicast (though this advantage over manual peer discovery will probably be negligible in most cases).

Configuring the HA Cluster for Manual Peer Discovery Support

Purpose

To modify your ehcache.xml configuration file for manual peer discovery support for your HA cluster.

Prerequisites

- You must have completely configured and installed your GIS nodes using the procedures found in [Installing and Uninstalling GIS](#).
- Your GIS nodes must be configured for HA deployment using [Configuring the GIS Application Objects](#).

Start

1. Confirm that an ehcache.xml configuration file is present in each GIS install directory (webapps\gis\conf\ for GIS:SOAP).
2. Configure each GIS instance using the following entries in the ehcache.xml file:

```
<cacheManagerPeerProviderFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"

properties="peerDiscovery=manual,rmiUrls=//localhost:40002/sampleDistributedCache1|
//localhost:40003/sampleDistributedCache1"/>
<cacheManagerPeerListenerFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"
properties="port=40001,socketTimeoutMillis=1000"/>
```

Important

GIS must find a cache element named sampleDistributedCache1. If this is not found, clustering will fail to work. If more cache configurations are found, they will be ignored.

Each GIS instance must be configured with:
a listening port, in this case 40001
the URLs to its peers, in this case:

```
//localhost:40002/sampleDistributedCache1
//localhost:40003/sampleDistributedCache1
```

Configuration of Three GIS Instances (Example) shows the configuration of three GIS instances.

Configuration of Three GIS Instances (Example)

GIS Application Name	Server Port	Peer URLs
GIS_SOAP	40001	//localhost:40002/ sampleDistributedCache1 //localhost:40003/ sampleDistributedCache1
GIS_SOAP2	40002	//localhost:40001/ sampleDistributedCache1 //localhost:40003/ sampleDistributedCache1
GIS_SOAP3	40003	//localhost:40001/ sampleDistributedCache1 //localhost:40002/ sampleDistributedCache1

End Next Steps

- Configure your client application by following the steps in [Configuring the Client Application](#).

Configuring the HA Cluster for Automatic Peer Discovery Support

Purpose

To modify your ehcache.xml configuration file for automatic peer discovery support for your HA cluster.

Prerequisites

- You must have completely configured and installed your GIS nodes using the procedures found in [Installing and Uninstalling GIS](#).
- Your GIS nodes must be configured for HA deployment using [Configuring the GIS Application Objects](#).
- You must have a multicast IP address in place.

Start

1. Confirm that an ehcache.xml configuration file is present in each GIS install directory (webapps\gis\conf\ for GIS:SOAP). This XML configuration file contains, by default, the following necessary cluster properties:

- Multicast heartbeat.
- Automatic node discovery.
- RMI communication between nodes.

2. Confirm the default contents of your ehcache.xml are:

```
<cacheManagerEventListenerFactory class="properties"/>
<cacheManagerPeerProviderFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
properties="peerDiscovery=automatic, multicastGroupAddress=230.0.0.1,
multicastGroupPort=4446"/>

<cacheManagerPeerListenerFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"
properties="<b>port=40001</b>, socketTimeoutMillis=2000"/>

<defaultCache
maxElementsInMemory="10000"
eternal="false"
timeToIdleSeconds="0"
timeToLiveSeconds="0"
overflowToDisk="true"
diskPersistent="false"
diskExpiryThreadIntervalSeconds="120"
memoryStoreEvictionPolicy="LRU"
/>

<cache name="sampleDistributedCache1"
maxElementsInMemory="10000"
eternal="false"
timeToIdleSeconds="0"
timeToLiveSeconds="0"
overflowToDisk="false">
<cacheEventListenerFactory
class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"/>
</cache>
```

The following table describes the important attributes from the ehcache.xml file.

Important ehcache.xml Attributes

Attribute	Description
peerDiscovery	Specify automatic peer discovery to allow the GIS node to be automatically recognized as a member of the cluster when starting.
multicastGroupAddress and multicastGroupPort	Specify the unique virtual address for the cluster. All GIS nodes in the cluster should be configured to use the same multicast group address and port.
port	Specify a unique node listening port . This value needs to

Attribute	Description
	be unique for each node that resides on the same host (although it can be the same for GIS nodes that are running on different machines).

End

Next Steps

- Configure your client application by following the steps in [Configuring the Client Application](#).

Configuring the Client Application

Purpose

To configure your GIS nodes so that your client application will automatically switch to the correct node when needed.

Prerequisites

- You must have completely configured and installed your GIS nodes using the procedures found in [Installing and Uninstalling GIS](#).
- Your GIS nodes must be configured for HA deployment using [Configuring the GIS Application Objects](#).
- Your ehcache.xml configuration file must be modified for your HA cluster using [Configuring the HA Cluster for Manual Peer Discovery Support](#).

Start

1. Refer to the *Agent Interaction SDK 7.6 Services Developer's Guide* for details about configuring the `Url` and `BackupUrls` options to configure the URLs of all GIS nodes.
2. For GIS:SOAP, using an HTTP Dispatcher, you must enable cookies in your client program. Ensure that the `UseCookieContainer` option is set to `true`.

Important

In the event that a T-Server or Interaction server to which GIS connects is recovering from a failover, GIS is not able to provide its own clients with the correct status for interactions with the status of `NEW` and `IDLE`. See the *Agent Interaction 7.6 Services Developer's Guide* for details on how to program for this in your client code.

End

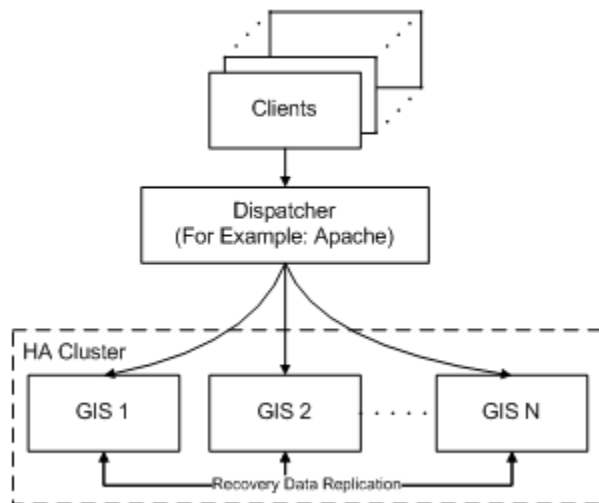
Next Steps

- To start and test GIS, [Starting and Testing GIS](#).

Deployment for High Availability with Load Balancing

You can deploy GIS with both HA and load balancing. To accomplish this, you must deploy an application server dispatcher (for instance, Apache) with your GIS:SOAP installation. The dispatcher takes client requests and parcels them out to available instances of GIS according to the load balancing rules you configure.

The following figure shows what the dispatcher deployment looks like. (Note that this is almost identical to the deployment without a dispatcher.) For this example, an instance of Apache is used with GIS:SOAP, which is served up by Tomcat.



Deployment for High Availability with Load Balancing

Generally, the configuration for standard GIS HA and HA plus load balancing are the same. A standard HA deployment with load balancing uses a cluster of GIS nodes, where each node is a GIS instance that has been fully configured and installed as described in:

- [Installing and Uninstalling GIS.](#)

Also, the procedure described in the [Configuring the HA Cluster for Manual Peer Discovery Support](#) must be completed.

In addition, you must complete the following 3 procedures for deploying GIS in HA mode with Load Balancing:

- [Configuring Apache Dispatcher](#)
- [Configuring Tomcat for Apache Dispatcher Deployment](#)
- See For GIS:SOAP, using an HTTP Dispatcher, you must enable cookies in your client program. Ensure that the UseCookieContainer option is set to true. of [Configuring the Client Application](#)

Important

The following procedures describe how to configure a load balanced HTTP cluster for SOAP.

Configuring Apache Dispatcher

Purpose

To configure the Apache Dispatcher to deploy two instances GIS with HA and Load Balancing.

Prerequisites

- You must have completely configured and installed your GIS nodes using the procedures found in [Installing and Uninstalling GIS](#).
- You must download the `mod_jk.so` Apache module matching your installed Apache version and place it in `Apache/modules/` directory.

Start

1. Include the following code in your Apache `httpd.conf` file. Ensure that the actual path to your `workers.properties` file is specified:

```
LoadModule jk_module modules/mod_jk.so
<IfModule mod_jk.c>
JkWorkersFile "C:\GCTI\Apache2\conf\workers.properties"
JkLogFile logs/jk.log
JkLogLevel error
JkMount /gis/ loadbalancer
JkMount /gis loadbalancer
JkMount /gis/* loadbalancer
</IfModule>
```

2. Create a `workers.properties` file, similar to the example provided below:

```
worker.list=gis1,gis2,loadbalancer
worker.gis1.port=8009
worker.gis1.host=host1
worker.gis1.type=ajp13
worker.gis1.lbfactor=1
worker.gis2.port=8009
worker.gis2.host=host2
worker.gis2.type=ajp13
worker.gis2.lbfactor=1
worker.loadbalancer.type=lb
worker.loadbalancer.balanced_workers=gis1, gis2
```

3. Refer to the Apache website for additional details on configuring load balancing.

Warning

If both GIS instances are installed on the same machine, be sure to assign a different AJP 1.3 connector port (configured in your `gis\conf\server.xml` file) to the second instance of GIS (specified as the `worker.gis2.port` value).

End Next Steps

- You must configure your Tomcat Application Server by following the steps in [Configuring Tomcat for Apache Dispatcher Deployment](#)

Configuring Tomcat for Apache Dispatcher Deployment

Purpose

To configure Tomcat for Apache Dispatcher.

Prerequisites

- You must have completely configured and installed your GIS nodes using the procedures found in [Installing and Uninstalling GIS](#).
- You must configure the Apache Dispatcher using the procedure [Configuring Apache Dispatcher](#).

Start

1. Modify your `gis\conf\server.xml` file, add `jvmRoute="gis1"` (where `gis1` is the worker name declared in your `apache workers.properties` file) in the following line:

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="gis1">
```

2. Locate the `<Connector port="8009" enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />` section in your `gis\conf\server.xml` file. Replace it with the following:

```
<Connector port="8009" maxThreads="1000" minSpareThreads="50"
maxSpareThreads="100" enableLookups="false" redirectPort="8443"
protocol="AJP/1.3" acceptCount="100" debug="0"
connectionTimeout="20000" disableUploadTimeout="true"/>
```

3. Locate and uncomment the following line in your `gis\webapps\gis\WEB-INF\web.xml` file:

```
<!-- <distributable>true</distributable> -->
```

4. Set the session value to `true` in this section of the `gis\webapps\gis\WEB-INF\web.xml` file, by adding the code below:

```
<servlet>
<servlet-name>GISAXIServlet</servlet-name>
<display-name>Apache-Axis Servlet</display-name>
<servlet-class>com.genesyslab.gis.framework.GISAXIServlet
</servlet-class>
<init-param>
<param-name>debug</param-name>
<param-value>>false</param-value>
</init-param>
```

```
<init-param>
<param-name>session</param-name>
<param-value>true</param-value>
</init-param>
<load-on-startup>2</load-on-startup>
</servlet>
```

End Next Steps

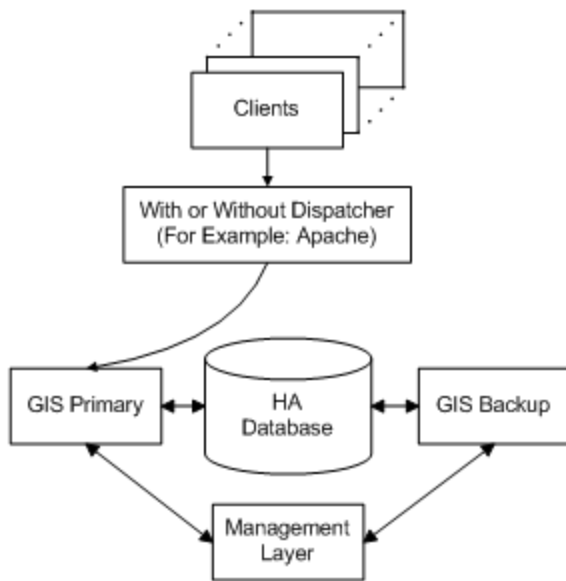
- Configure your client application according to See For GIS:SOAP, using an HTTP Dispatcher, you must enable cookies in your client program. Ensure that the UseCookieContainer option is set to true. of [Configuring the Client Application](#).

Important

With the dispatcher and load balancing, your client can use a single URL (Url option) for connection to the GIS cluster. This will remain the same regardless of any failover activity.

Alternative Deployment for High Availability

GIS offers a HA deployment alternative. This deployment is designed for Primary/Backup GIS pairs. It relies on the Genesys Management Layer (LCA) Primary/Backup monitoring mechanism, requires the use of a database as a third-party recovery-data storage unit, and can be implemented with SOAP. Your client applications automatically connect to the backup GIS instance if the primary instance fails. The Management Layer (LCA) controls this failover process, and the backup GIS instance uses sessions data stored in the (third-party) recovery database to recover smoothly from the failover.



Alternative High Availability Deployment

This HA deployment uses two GIS instances that have been fully configured and installed as described in:

- [Configuring and Installing GIS.](#)
- [Installing and Uninstalling GIS.](#)

There are 3 main procedures that must be followed to deploy the Primary/Backup GIS pair in HA mode:

- [Configuring the GIS Application Objects](#)
- [Configuring the HA Cluster for Manual Peer Discovery Support](#)
- [Configuring the Client Application](#)

Configuring and Installing GIS

Purpose

To create a GIS HA database and a pair of GIS applications (along with supporting configuration objects) that form a Primary/Backup pair.

Prerequisites

- Review the database options found in [System Requirements](#).
- Ensure that you have the required templates. Refer to the Genesys Framework Deployment Guide for information on how to import a template into Configuration Manager.

Important

One database and one Data Access Point (DAP) is required for each Primary/Backup GIS pair.

Start

1. Create a GIS HA database to hold recovery information for the backup server. You can use the Genesys Universal Contact Server (UCS) database for this purpose, or you can use a Microsoft SQL, DB2, or Oracle database. To prevent conflicts, high-availability tables in that database have names that begin with a prefix of `ha_`.
2. In Configuration Manager, configure a DAP for the database you created in Step 1 above. When creating this DAP, select JDBC connectivity.
3. Create and configure a homogeneous pair of GIS Application objects.

Important

Each GIS instance should connect to a distinct Agent Interaction Service Application object and (if required) Open Media Interaction Service Application object.

4. In your primary GIS application, configure connections to your DAP, Agent Interaction Service, and (if required) Open Media Interaction Service.
5. On your primary GIS application's **Server Info** tab, set the **Backup Server** field to the second GIS application in your pair. (This automatically duplicates the connections that you configured in See In your primary GIS application, configure connections to your DAP, Agent Interaction Service, and (if required) Open Media Interaction Service. above.) Set the **Redundancy Type** field to **Hot Standby**.

Important

Despite the name of this selection, GIS currently employs a warm-standby approach to high availability.

6. Repeat See On your primary GIS application's **Server Info** tab, set the **Backup Server** field to the second GIS application in your pair. (This automatically duplicates the connections that you configured in Step 4 above.) Set the **Redundancy Type** field to **Hot Standby**. for the primary Agent Interaction Service and Open Media Interaction Service Application objects.
7. On your backup GIS application's **Server Info** tab, leave the **Backup Server** field empty. Set the **Redundancy Type** field to **Hot Standby**.
8. Install two homogeneous instances of GIS, as described in the appropriate section from **Installing and Uninstalling GIS**:
The GIS installer includes the drivers required to access your database engine, and automatically installs them in the required location:
 - **MS SQL:** `jtds-1.1.jar`

- **DB2:** db2jcc.jar
- **Oracle:** ojdbc14.jar

End Next Steps

- Configure the server side according to [Configuring the Server Side](#).

Configuring the Server Side

Purpose

To configure the dispatcher to support high availability servers.

Prerequisites

- Complete the procedure [Configuring and Installing GIS](#).
- You must be planning to use a dispatcher (See [Deployment for High Availability with Load Balancing](#)).

Start

Important

This procedure assumes that you are using an Apache server as the dispatcher.

1. Configure your `workers.properties` file as follows:

```
worker.list=gis1,gis2,loadbalancer
worker.gis1.port=8009
worker.gis1.host=host1
worker.gis1.type=ajp13
worker.gis1.lbfactor=1
# Define preferred failover node for worker1
worker.gis1.redirect=gis2
worker.gis2.port=8009
worker.gis2.host=host2
worker.gis2.type=ajp13
worker.gis2.lbfactor=1
# Disable gis2 for all requests except failover
worker.gis2.disabled=True
worker.loadbalancer.type=lb
worker.loadbalancer.balanced_workers=gis1, gis2
```

Warning

If both GIS instances are installed on the same machine, assign a different port number to the second instance (the `worker.gis2.port` value).

End Next Steps

- Configure the client side using the procedure that suits your deployment type:
 - [Configuring the Client Side with Dispatcher](#).
 - [Configuring the Client Side without Dispatcher](#)

Configuring the Client Side with Dispatcher

Purpose

To configure the client side with a dispatcher.

Prerequisites

- Complete the procedure [Configuring and Installing GIS](#).
- Complete the procedure [Configuring the Server Side](#).

Start

1. Configure your dispatcher, as described in [Configuring Apache Dispatcher](#).

Important

The client application needs to manage only one URL, so it requires no specific coding to incorporate high-availability support.

End Next Steps

- Start GIS, [Starting and Testing GIS](#)

Configuring the Client Side without Dispatcher

Purpose

To configure the client side without dispatcher.

Prerequisites

- Complete the procedure [Configuring and Installing GIS](#).

Start

1. Configure the `Url` and `BackupUrls` options according to the *Agent Interaction SDK 7.6 Services Developer's Guide*.

End Next Steps

- Start GIS, see [Starting and Testing GIS](#).

Starting and Testing GIS

The final step in the installation and configuration process is to confirm that GIS is properly installed and runs successfully. This chapter includes the following sections:

- [Starting GIS](#)
- [Stopping GIS](#)

Starting GIS

If you used the default installation directory, the start scripts are already set up so that `/GIS/bin/` is the current directory.

Choosing Your Start Script

Purpose

To choose the correct start script to startup GIS.

Prerequisites

- Check that you have set the `JAVA_HOME` environment variable correctly. For details, see [Setting the JAVA_HOME Environment Variable](#).

Start

1. Locate the start scripts:
 - For GIS:SOAP installations, the start scripts are located in the `bin/` subdirectory under the top-level installation directory.
2. Select the start script for your environment platform:
 - For GIS:SOAP installations:
 - `run_gis_soap.sh` is a UNIX Bourne shell script compatible with Korn and with BASH shells. It is used for startup on Solaris, HP-UX, Linux and AIX platforms.
 - `run_gis_soap.bat` is a batch-file script used for startup on Windows platforms.
3. Verify that the values, entered during the IP installation, are correct by using a text editor to open and inspect the start script. To do so, follow the steps in the [Editing the Start Script](#) procedure.
4. Run the script.

End

Next Steps

- You may need to modify the start script by following [Editing the Start Script](#).

Editing the Start Script

Purpose

To verify that the values, entered during the IP installation, are correct by using a text editor to open and inspect the start script.

Prerequisites

- You must have selected a start script based on your needs, see [Choosing Your Start Script](#).

Start

1. Use a text editor to open and inspect the Windows, AIX, HP-UX, Linux or Solaris start script (whichever version you intend to use) to verify that the values, entered during the IP installation, are correct.
2. Edit the value of the port parameter found in the `.../conf/server.xml` file, located under the top-level installation directory to change the port for Tomcat. The default port assignment for Tomcat is 8080. If you change this port number, any references to port 8080 in the *Statistics SDK* and *Configuration SDK Developer's Guides* and *API References* must refer to the port number you configured here.

End

Next Steps

- Start GIS, [Starting and Testing GIS](#).

Starting and Testing GIS

Purpose

To start and test GIS to confirm installation and configuration is correct.

Prerequisites

- Confirm that Configuration Server is running.

Start

1. Run the correct start script to start up GIS. See [Choosing Your Start Script](#).
2. Access GIS using a web browser:
3. Follow one of the start procedures:
 - [Starting on UNIX-based Platforms \(GIS:SOAP\)](#).
 - [Starting GIS on Windows](#).
 - [Starting GIS on a Web Application Server](#).

End

Next Steps

- For GIS:SOAP, you may want to specify a licensing at startup, [Specifying Licensing at Startup \(GIS:SOAP\)](#).

Configuring the License Section of the Option Tab for GIS SOAP

Purpose

To configure the [license Section](#) of the Options tab for GIS:SOAP during configuration of a GIS Application object.

Important

You can also configure the license file parameter at startup, see [Specifying Licensing at Startup \(GIS:SOAP\)](#).

Prerequisites

- You must have configured your application for GIS:SOAP

Important

GIS does not support redundant License Managers. However, you can specify multiple independent License Managers, using one of the following formats:

- Windows: server1;server2;server3 [and so on].
- UNIX: server1:server2:server3 [and so on].

You can also enter the license information, in either format, on the command line at startup, see [Specifying Licensing at Startup \(GIS:SOAP\)](#). If you do so, the command-line information takes priority over the [license-file](#) option settings. See [Starting and Testing GIS](#), for details.

Important

If you are upgrading to GIS 7.6 from Genesys Interface Server 7.1 or earlier, migrate your existing license keys. For details, see the [Genesys Licensing Guide](#).

Start

1. Double-click license or select it from the Sections combo box of the Options tab of the GIS Application object.
2. Open the [license-file](#) option.
3. For the license-file option value, enter either:
 - The value <port>@<host> for the host running the FlexLM License Manager daemon.
 - The path to a local license file.
4. Use the information found in the [license Section](#) as a guide to configuring additional settings in the

Options tab's license section.

End Next Steps

- Configure other sections of the Options Tab, see [Configuration Options](#).

Specifying Licensing at Startup (GIS SOAP)

Purpose

To configure the location of the license file using a command line.

Prerequisites

- You must have configured your application as a SOAP server.

Important

You can also configure the license file location during configuration of the GIS Application object (see [Genesys Integration Server](#)).

Start

1. Append the `-l` option on the command line, followed by either:
 - The License Manager `<port>@<host>` argument to specify the port number and host name for the License Manager server.
 - The full path to the license file, including the license file name.

Important

If you specify license information in both the command-line and

the during the configuration of the GIS Application object, the command-line option takes priority over the `license-file` option set in Configuration Manager. GIS does not support redundant License Managers. However, you can specify multiple independent License Managers, using one of the following formats.

- For Windows: `-l "server1;server2;server3"` [and so on—include the quotes].
- For UNIX-based: `-l server1:server2:server3` [and so on].

End Next Steps

- [Verify the Status of GIS \(GIS:SOAP Only\)](#).

Starting on UNIX-based Platforms (GIS SOAP)

Purpose

To start GIS:SOAP on UNIX-based platforms.

Prerequisites

- You must have configured your application as a SOAP server.

Start

1. Open a shell window.
2. At the command line, make your GIS installation directory the current directory (for example: `cd /GIS`).
3. At the command line, enter one of the following:
 - `gis.sh -l <lic_server_port>@<lic_server_host> -host <cfg_host> -port <cfg_port> -app <app_name>`.
 - `./bin/run_gis_soap.sh` (for GIS:SOAP installations).

Important

If you use the second command, specify your `<lic_server_port>@<lic_server_host>` licensing information in the `run_gis_soap.sh` or `startServer.sh` file.

End

Next Steps

- [Verify the Status of GIS \(GIS:SOAP Only\)](#).
- [Stopping GIS on UNIX-based Platforms](#).

Verify the Status of GIS (GIS SOAP)

Purpose

To check that a GIS:SOAP instance is running properly.

Prerequisites

- [Starting on UNIX-based Platforms \(GIS:SOAP only\)](#).

Start

1. Open a web browser.
2. Open the GIS web page, as follows:

- If you installed GIS as a stand-alone server, enter the following URL in the address bar:
`http://<gis_host>:8080/gis`
- If you installed GIS in WebSphere Base 5.x, enter the following URL in the address bar:
`http://<WebSphere_host>:9080/gis76`
- If you installed GIS in WebSphere Express 5.x, enter the following URL in the address bar:
`http://<WebSphere_host>:7080/gis76`
- If you installed GIS in WebSphere 6.0.x, enter the following URL in the address bar:
`http://<WebSphere_host>:9081/gis76`

Important

The port numbers listed above are the default ports for each type of GIS installation. If you changed the port during configuration and installation, adjust the URL accordingly.

3. Confirm that browser has connected to GIS web page. The page shows the following links:
 - **Overview**—Presents basic statistics such as the GIS version number, Java and system resources, and essential server information.
 - **SessionService**—Presents information about service state and number of sessions.
 - **StatService**—Presents connectivity information about service state, license name, Stat Server, and current number of statistics subscribed to by agents.
 - **CSProxyService**—Presents information about service state and license name, the CS Proxy SOAP interface, and configuration information. Also enables changes to configuration objects.
 - **AIL**—Presents connectivity information about service state, license name, and the AIL Server.
 - **OPENMEDIA**—Presents connectivity information about service state and the Open Media Server.

End Next Steps

- [Stopping GIS on UNIX-based Platforms.](#)

Starting GIS on Windows

Purpose

To start GIS on Windows platforms, use either a Start menu shortcut or a batch file.

Prerequisites

- You must have configured your application to be ran on a Windows platform.

Start

Using the Start menu:

1. Select Start > Programs > Genesys Solutions > Genesys Integration Server > > <protocol> > Start Server.

Using the batch file (for GIS:SOAP):

1. Navigate to the <GIS_installation_directory>/bin/ subdirectory.
2. Double-click run_gis_soap.bat.

End Next Steps

- [Stopping GIS on Windows.](#)

Starting GIS on a Web Application Server

Purpose

To start GIS on a web application server such as WebSphere or WebLogic, consult your application server's documentation.

Prerequisites

- You must have deployed your application on WebSphere or WebLogic.

Start

1. Locate and edit the gis.properties file to modify the startup parameters, such as the host name or port number:
 - For WebSphere, this file is located in the ../AppServer/installedApps/<node_name>/<application_name> /gis.war/conf/ directory under your WebSphere directory structure.
 - For WebLogic, this file is located in the <gis_deployment_directory>/gis_72/conf/ subdirectory.

End Next Steps

- [Stopping on a Web Application Server.](#)

Stopping GIS

If you are not using the Management Layer, stop GIS using the appropriate method as described in this section.

Stopping GIS on UNIX-based Platforms

Purpose

To stop GIS on UNIX-based platforms.

Prerequisites

- You must be running GIS on a UNIX-based platform.

Start

1. Open a shell window.
2. At the command line, change the current directory, using the command appropriate for your installation type:
 - (For GIS:SOAP:) `cd <GIS_installation_directory>/bin`
3. Issue the `./shutdown.sh` command (for GIS:SOAP).

End

Stopping GIS on Windows

Purpose

To stop GIS on Windows platforms by using either a Start menu shortcut or a batch file.

Prerequisites

- You must be running GIS on a Windows platform.

Start

Using the Start menu:

1. Select Start > Programs > Genesys Solutions > Genesys Integration Server > <protocol> > Stop Server.

Using the batch file (for SOAP):

1. Navigate to the <GIS_installation_directory>/bin directory.
2. Double-click `shutdown.bat`.

End

Stopping on a Web Application Server

Purpose

To stop GIS on a web application server such as WebSphere or WebLogic.

Prerequisites

- You must be running GIS on a on a web application server such as WebSphere or WebLogic.

Start

1. Consult your application server's documentation.

End

Using Management Layer to Start and Stop GIS

Purpose

To start/stop GIS from the Solution Control Interface (SCI).

Prerequisites

- GIS must be installed.

Important

You cannot use the Management Layer to control GIS if you are running GIS in a WebSphere or WebLogic web container.

Start

To start GIS through SCI:

1. Start the SCI.
2. Go to the Applications view.
3. Right-click the GIS application and select Start from the shortcut menu.**-or-** Select the GIS application and choose Action > Start on the menu bar.

To stop GIS through SCI:

1. Start the SCI.
2. Go to the Applications view.
3. Right-click the GIS application and select Stop from the shortcut menu.**-or-** Select the GIS application and choose Action > Stop on the menu bar.

End

Configuration Options

This appendix includes information on configuring options in the Application objects for Agent Interaction Service, Open Media Interaction Service, and Genesys Integration Server. In Configuration Manager, configuration options are stored within *section* folders. The section folders and options are stored on the Options tab; you click a section to bring up its options. Options are described in detail the following pages.

- Agent Interaction Service Options
- Open Media Service Options
- Genesys Integration Server Options

Agent Interaction Service Options

You must configure the following before starting the Agent Interaction Service:

- The location or address of the FlexLM license under the `license` option. For details about the license settings, and a list of other available options, see the [license Section](#).
- (Optional) The `email-default-queue` option in the [multimedia Section](#) for e-mail interactions, if you are using the .NET Toolkit to send or receive e-mail.

dn-at-switch Section

enabled

- Default Value: `false`
- Valid Values: `true`, `false`
- **Description:** Used when there are several switches declared in the same configuration. For example, the switch DN ID becomes `103@Xswitch`.

kworker Section

auto-markdone

- Default Value: `true`
- Valid Values: `true`, `false`
- **Description:** If set to `true`, an interaction on T-Server without CTI is automatically marked done when it is released.

easy-newcall

- Default Value: `true`
- Valid Values: `true`, `false`
- **Description:** If set to `true`, a newly created interaction is automatically dialed.

reroute

- Default Value: false
- Valid Values: true, false
- **Description:** Specifies whether the reroute capability is enabled (true) or disabled (false).

reroute-location

- Default Value: <location>
- Valid Values: switch1,...,switchN
- **Description:** Specifies which switch locations can receive rerouted calls. This option is used only if the **reroute** option is set to true. If set to the default value of <location>, calls can be rerouted to any switch. If set to another valid value, rerouting is restricted to the specified switches.

license Section

attempts-interval

- Default Value: 5 (in seconds)
- Valid Values: <Any positive integer>
- **Description:** Time interval, in seconds, between two successive connection attempts.

attempts-max

- Default Value:10
- Valid Value: <Any positive integer>
- **Description:** Maximum number of successive connection attempts to the server before triggering an exception.

license-file

- Default Value: license.dat
- Valid Values for Windows:
 - <licenseserver_port1@hostname1>;<licenseserver_port2@hostname2>
 - **Description:** A semi-colon-separated list of addresses of Flexlm license servers or explicit paths to

the actual license file.

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

loading Section

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

srl-on-demand

- Default Value: false
- Valid Values: true, false
- **Description:** Specifies the method to load SRL (Standard Response List) objects, allowing you to optimize your application.
 - If set to true: SRL trees are loaded on demand.
 - If set to false: SRL trees are loaded on startup.

log Section

Miscellaneous Traces.

console

- Default Value: info
- Valid Values: false, debug, info, warn, error, fatal
- **Description:** Level and size of traces to display on the standard output.

file

- Default Value: info, ail, 10MB, 20, zip
- Valid Values: <level>, <file_name_root>, <file_max_size>, <file_number> [, zip] [, timestamped]
 - <level>: false, debug, info, warn, error, fatal
 - <file_name_root>: correct path to a file name

- `< file_max_size>`: maximum file size in MB
- `< file_number>`: number of files for the rolling logs
- `[, zip]`: to get compressed log files (optional)
- `[, timestamped]`: to add a timestamp when a new file is created. The existing file is replaced when it reaches `file_max_size` (optional)
- **Description:** Specifies how to write entries in log files.

multimedia Section

chat-addp-protocol

- Default Value: `false`
- Valid Values: `true`, `false`
- **Description:** To use the ADDP protocol to connect to the chat server, set this option to `true`.

chat-busy-threshold

- Default Value: `1`
- Valid Value: `<Any integer>`
- **Description:** Threshold number of chat interactions beyond which an agent is seen as busy.

Important

Not connected to routing strategy.

collaboration-workbin

- Default Value: `desktop-collaboration-email-workbin`
- Valid Values: `<Any string value>`
- **Description:** Used for desktop collaboration feature. When inviting an agent in `pull` mode, the system stores the internal invitation in the specified agent workbin.

email-address-rfc822-strict

- Default Value: `false`

- Valid Values: true, false
- **Description:** If set to true, AIL checks if the e-mail addresses of an interaction are compliant with the RFC-822 Standard for the format of ARPA Internet text messages.

email-default-queue

- Default Value: N/A
- Valid Values: *<Any string value>*
- **Description:** Specifies a valid queue for the AIL Application object. You must add this option to the multimedia Section before sending or receiving e-mail interactions using the .NET Toolkit.

email-busy-threshold

- Default Value: 1
- Valid Values: *<Any integer>*
- **Description:** Threshold number of e-mail interactions beyond which an agent is seen as busy.

Important

Not connected to routing strategy.

email-quote-char

- Default Value: >
- Valid Values: *<Any string value>*
- **Description:** When the agent replies to an e-mail, this string precedes each line of the sender's original message.

email-quote-header

- Default Value: On <date>,<contact> wrote:
- Valid Values: *<Any string value>*
- **Description:** Specifies the header that precedes the sender's original message. The header can contain two dynamic values:<date> and <contact>.

enable-multicharset-environment

- Default Value: false
- Valid Values: true, false
- Changes Take Effect: After restart.
- **Description:**
 - If set to true, unicode data will be retrieved from Contact Server.
 - If set to false, non-unicode data will be retrieved from Interaction Server.

open-media-saved-list

- Default Value: none
- Valid Values: Media types separated by commas
- **Description:** A comma-separated list of valid media types that will be saved in UCS, such as sms and video . Valid media types are created and configured in the Business directory of Configuration Manager.

logout-multimedia-on-session-removal

- Default Value: false
- Valid Values: true, false
- **Description:** If set to true, forces the multimedia logout on session removal regardless of the object reference. Logout will be done only if a multimedia login has occurred in this session. All media will be logged out.

outbound Section

enable-chain-75api

- Default Value: true
- Valid Values: true , false
- Changes Take Effect: After restart.
- **Description:** If set to true, enables the use of the new API with the OutboundChain class, and abandons the InteractionVoiceOutbound class.

settings Section

ucsapi-max-active-rmi-client

- Default Values: 40
- Valid values: *<Any integer value>*
- **Description:** Maximum number of active clients that are able to connect to ucsapi .

ucsapi-max-timeout-rmi-client

- Default Values: 30
- Valid values: *<Any integer value>*
- **Description:** The time, in seconds, that additional clients will wait before the request times out.

signature Section

include-agent-name

- Default Value: true
- Valid Values: true, false
- **Description:** If true, add the agent's name at the begin of the signature of an e-mail, after the prefix.

line1, line2....lineN

- Default Value: " "
- Valid Values: *<Any string value>*
- **Description:** Use to define each line of the signature.

prefix

- Default Value: " - "
- Valid Values: *<Any string>*, *<empty string>*
- **Description:** Use this option to set a separator string before your signature. If this option is set to

<empty string>, there is no additional line in the standard signature.

voice Section

a4400-custom-substitute-mode

- Default Value: true
- Valid Values: true, false
- Changes Take Effect: After restart.
- **Description:** Use this option to customize substitute behavior. When set to true, the virtual position DN that is created when an agent logs in is replaced with the extension DN associated with the Place.

database

- Default Value: all
- Valid Values: all, external, manual, none
- Changes Take Effect: Immediately.
- **Description:** This option specifies the use of the UCS database for voice calls when Contact Server DB is connected.
 - all—Any voice call uses the database.
 - external—Internal calls do not use the database.
 - manual—AIL no longer manages the automatic contact lookup and the creation of voice interactions in the UCS database. In this mode, your application is responsible for the following actions:
 - Fetching (or creating) the ContactServerId DBID for each contact
 - Attaching the ContactServerId DBID to the voice interaction
 - Saving contact records and interactions in the UCS database
 - none—No voice call uses the database.

dms-last-digits

- Default Value: -1
- Valid Values: *<Any positive integer>*
- **Description:** For Nortel Communication Server 2000 (DMS 100) switch only. Specifies how many digits should be retained at the end of a DN number to get its dialable number. For example, if the DN number is 1001234567 and this option is set to 4, this DN is called from 4567. If the value is -1 or if the

resulting transformation does not provide a correct number, the system uses the former number.

enable-all-routing-events

- Default Value: false
- Valid Values: true, false
- **Description:** Option to send all events to `RoutingInteractionListeners`. Default is false (to receive only NEW, IDLE, and INFO-CHANGES events). Set to true to receive all the events.

enable-attached-data-for-transfer

- Default Value: true
- Valid Values: true, false
- **Description:** Use this option to disable the attachment of GD_* data when transferring a phone call.

enable-interaction-id-tracking

- Default Value: true
- Valid Values: true, false
- **Description:** If set to true, enables every means necessary to maintain uniqueness of interaction IDs.

enable-possible-changed-event

- Default Value: true
- Valid Values: true, false
- **Description:** Determines an interaction's behavior when its possible actions have changed. If set to true, an interaction sends an `InteractionEvent`, together with the current interaction status and the `POSSIBLE_CHANGED` event reason, if the interaction's possible actions have changed due to the status change of another related interaction (consult, primary). If set to false, you will not receive those events.

logout-voice-on-session-removal

- Default Value: false
- Valid Values: true, false
- **Description:** If set to true, forces the voice logout on session removal regardless of the object reference. The logout will be done only if a voice login has been performed in this session. All DNs will

be logged out.

idle-interactions-cleanup-delay

- Default Value: -1
- Valid Values: <Any integer>
- Changes Take Effect: Immediately.
- **Description:** The length of time, in minutes, that an interaction can be IDLE before it can be silently removed when one or more further interactions become IDLE. This option is intended to prevent memory leaks by removing IDLE interactions in AIL. Set to -1 to disable this option.

Open Media Service Options

This section describes the configuration options for the Open Media Interaction Service Application object. To set various configuration options for the application, use the Options tab of the Properties dialog box and use the following information as a guide.

esp Section

esp-response-timeout

- Default Value: 2000
- Valid Values: Integer values from 100 to 10000
- **Description:** Specifies the response timeout, in milliseconds, for the External Service Property (ESP) service. If the client's ESP handler does not handle a request within the specified period, MIL will send the fault message as a response.

general Section

agent-place

- Default Value: [none]
- Valid Values: A string representing an existing system Place (a Place that is not used by an Agent).
- **Description:** Used to connect to the Interaction Server to be able to perform ad-hoc features.

period

- Default Value: 60
- Valid Values: A valid integer.
- **Description:** Seconds to wait before a reconnection attempt. This option applies to all the QIL and MIL back-end connections after a GIS restart.

reconnectionAttempts

- Default Value: 1

- Valid Values: A valid integer.
- **Description:** Number of reconnections to attempt in case of connection or disconnection. This option applies to all the QIL and MIL back-end connections after a GIS restart.

log Section

omsdk-console

- Default Value: info
- Valid Values: false, debug, info, warn, error, fatal
- **Description:** Level and size of traces to display on the standard output.

omsdk-file

- Default Value: info, mil.log, 10, 20
- Valid Values: `<level>`, `<file_name_root>`, `<file_max_size>`, `<file_number>` [, zip]
 - `<level>`: false, debug, info, warn, error, fatal
 - `<file_name_root>`: A valid path to a file name.
 - `<file_max_size>`: Maximum file size, in MB.
 - `<file_number>`: Number of files for the rolling logs.
 - [, zip]: Specifies compressed log files (optional).
- **Description:** Specifies how to store traces to a file.

omsdk-msgsrv

- Default Value: off
- Valid Values: off, debug, info, warn, fatal
- **Description:** Specifies level and size of this server's logging traces for the MessageServer centralized log. If this option is not specified, this server does not log to MessageServer.

server Section

type

- Default Value: openmedia
- Valid Values: openmedia

Warning

Do not change this value. Changes made to the type option of the server Section are not supported.

settings Section

enable-attached-data-byte-array

- Default Value: true
- Valid Values: true, false
- **Description:** Specifies the conversion method for binary data within ESP (External Service Property) requests and responses.
 - If set to true, then MIL (the Media Interaction Layer) converts binary data from requests to an array of bytes (byte[]). MIL also expects that binary data in responses will be of the same type.
 - If set to false, the MIL converts binary data to an ArrayList of objects with class Byte. MIL also expects that binary data in responses will be of the same type.

use-esp

- Default Value: true
- Valid Values: true, false
- **Description:** When set to true, specifies that you use a low-level ESP service in MIL.

use-lca

- Default Value: false

- Valid Values: false
- **Description:** If set to true , specifies that you use a low-level LCA service in MIL.

Warning

Do not change this value. Changes are not supported.

ucs Section

ucs-identify-create-contact

- Default Value: false
- Valid Values: true, false
- **Description:** Triggers the UCS (Universal Contact Server) service's create () method. If this option is set to true , it directs UCS to create a new contact when a contact is not found, and to create a corresponding interaction.

workflow Section

request-timeout

- Default Value: 50000
- Valid Values: Integer values from 100 to 10000.
- **Description:** Specifies the request timeout for the Interaction Server service. If Interaction Server does not respond within the specified period (in milliseconds), then Open Media Server will throw a corresponding TimeoutException.

Genesys Integration Server Options

You must configure the following GIS Application object configuration options before starting GIS:

- On the Options tab, specify the location or address of the FlexLM license file, see [Configuring the License Section of the Option Tab \(for GIS:SOAP\)](#).

Further, the GIS Application object can be customized, with options available for the Session service, Statistics service, Core service, Open Media Interaction service, and Agent Interaction service, as described in the following sections:

- [ail-services Section](#)
- [core-services Section](#)
- [license Section](#)
- [log Section](#)
- [session-services Section](#)
- [stat-services Section](#)

ail-services Section

agent-wait-status-stable

- Default Value: 100
- Valid Values: Integer values from 0 to 1000.
- Changes Take Effect: After restart.
- **Description:** Time, in milliseconds, to wait for the agent to complete the login action. If set to 0, do not wait at all.

cache-lease-time

- Default Value: 30
- Valid Values: Integer values.
- Changes Take Effect: After restart.
- **Description:** Time (in minutes) that the objects DN, Agent, and Place remain in the internal cache. If set to 0, the objects are not cached.

interaction-voice-create-new-timetolive

- Default Value: 30
- Valid Values: Integer values from 0 to 60.
- Changes Take Effect: After restart.
- **Description:** Maximum time (in minutes) to maintain an interaction in the status NEW, following a call to the `createInteraction()` method. If this time period expires, then the interaction is removed.

period

- Default Value: 60
- Valid Values: A valid integer.
- **Description:** Seconds to wait before a reconnection attempt. This option applies to all the AIL back-end connections after a GIS restart.

receive-self-user-event

- Default Value: false
- Valid Values: true, false
- Changes Take Effect: Immediately.
- **Description:** If set to true, GIS will send to its clients the same UserEvent message that it sent to the T-Server.

reconnectionAttempts

- Default Value: 1
- Valid Values: A valid integer.
- **Description:** Number of reconnections to attempt in case of connection or disconnection. This option applies to all the AIL back-end connections after a GIS restart.

timeout

- Default Value: 30
-

- Valid Values: Any positive integer.
- Changes Take Effect: After restart.
- **Description:** Specifies the timeout value, in seconds, of requests sent to back-end servers. For example, loading cfg data.

core-services Section

events-adapter-soap-commons-sender-enabled

- Default Value: true
- Valid Values: true, false
- Changes Take Effect: Immediately.
- **Description:** Enables the commons sender HTTP adapter, using SOAP (Simple Object Access Protocol) over HTTP 1.0. This is the default adapter.

events-adapter-soap-commons-sender-http11-mode

- Default Value: true
- Valid Values: true, false
- Changes Take Effect: Immediately.
- **Description:** Enables HTTP 1.1 mode for the commons sender HTTP adapter.

events-adapter-soap-commons-sender-max-connections-host

- Default Value: 50
- Valid Values: Integer values from 1 to 50.
- Changes Take Effect: Immediately.
- **Description:** Sets the maximum of connections, per host, that the commons sender HTTP adapter uses.

events-adapter-soap-commons-sender-max-total-connections

- Default Value: 500

- Valid Values: Integer values from 5 to 500.
- Changes Take Effect: Immediately.
- **Description:** Sets the maximum of connections that the commons sender HTTP adapter uses.

events-adapter-soap-invokeonewaymode

- Default Value: false
- Valid Values: true, false
- Changes Take Effect: Immediately.
- **Description:**
 - If true, calls the `notifyEvents()` method on the subscriber in one-way mode.
 - If false, calls this method in requestresponse mode. Applies only in push mode.

events-adapter-soap-notification-failure-nbretries

- Default Value: 3
- Valid Values: Integer values from 0 to 5.
- Changes Take Effect: Immediately.
- **Description:** Number of retries upon notification failure. If this threshold is exceeded, the subscriber is removed. Applies only in push mode.

events-adapter-soap-notification-polling

- Default Value: true
- Valid Values: true, false
- Changes Take Effect: Immediately.
- **Description:** If true, calls the `notifyEvents()` method on each subscriber that has no events, to verify that the subscriber is still present. The call is repeated at intervals specified by the `events-subscriber-timetolive` polling option's value. If a call fails, the subscriber is removed. Applies only in push mode.

events-buffered-period

- Default Value: 100
- Valid Values: Integer values from 0 to 1000.

- Changes Take Effect: Immediately.
- **Description:** Period, in milliseconds, by which events are buffered before they are sent to subscribers (or subscribers retrieve them).
If this option is set to 0, the events are not buffered.

events-buffered-size

- Default Value: 200
- Valid Values: Integer values.
- Changes Take Effect: Immediately.
- **Description:** Maximum number of events buffered before they are sent to subscribers (or subscribers retrieve them). If this option is set to 0, events are sent immediately. If the buffer overflows, the oldest events are removed.

events-keep-alive-time

- Default Value: 5
- Valid Values: Integer values from 1 to 60 .
- Changes Take Effect: After restart.
- **Description:** Number of minutes to keep threads alive while they are waiting in the notification pool.

events-maximum-pool-size

- Default Value: 100
- Valid Values: Integer values from 20 to 100.
- Changes Take Effect: After restart.
- **Description:** Maximum number of threads in the notification pool.

events-minimum-pool-size

- Default Value: 10
- Valid Values: Integer values from 0 to 20.
- Changes Take Effect: After restart.
- **Description:** Minimum number of threads in the notification pool.

events-subscriber-timetolive-polling

- Default Value: 10
- Valid Values: Integer values from 5 to 60.
- Changes Take Effect: Immediately.
- **Description:** Time limit, in minutes, within which the subscriber must call the `getEvents()` method. If there is no call to this method, the subscriber is removed. Applies in both push and polling modes.

Important

By setting `session-cache-enabled` to `true`, you also prevent subscribers in pull mode from being automatically removed by the `events-subscriber-timetolive-polling` timer option.

events-thread-priority

- Default Value: 7
- Valid Values: Integer values from 1 to 10.
- Changes Take Effect: After restart.
- **Description:** Notification threads' priority (where 1 represents minimum, 5 represents normal, and 10 represents maximum). Applies only in notification mode.

ha-died-recovering

- Default Value: `true`
- Valid Values: `true`, `false`
- Changes Take Effect: Immediately.
- **Description:** Enables automatic recovery when a dead node is detected in the cluster. If `false`, the cluster will wait until the next client request before recovering. Applicable only to cluster based HA, in `events-push` mode.

session-cache-enabled

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

- Changes Take Effect: Immediately.
- In GIS, cache lease time of objects is activated by default (30 minutes in the CME options tab). When working with multimedia, the cache lease time option automatically logs out the user (and removes any event subscription) on inactivity when the timeout is reached, even if the user session timeout has not expired.
To avoid an inappropriate logout in multimedia server(s), setting the session-cache-enabled option to true links the lease time to the user session. As long as the user session is alive, lease time will not be involved. If the user session is no longer found, the lease time is evaluated.
By setting session-cache-enabled to true, you also prevent subscribers in pull mode from being automatically removed by the events-subscriber-timetolive-polling timer option.

Important

Lease time behavior only applies with multimedia, not voice. Therefore, if the user is only doing voice activity, setting this option will not have any affect on GIS behavior. If the user is doing multimedia activity, set this option to true.
A negative lease time value never removes objects from cache. A null value forces GIS to refresh objects each time. Objects are not kept in cache and activates the logout mechanism described.

license Section

gis_interactionservice-block-size

- Default Value: 1
- Valid Values: Any positive value.
- **Description:** Specifies the number of interaction corporate licenses for connections to FlexLM server. The valid value must not exceed the number of licenses that your company has purchased.

gis_interactionservice-firstblock-size

- Default Value: 0 (no specific first request)
- Valid Values: Any positive value.
- **Description:** Specifies the number of interaction corporate licenses for the first request to FlexLM server. The valid value must not exceed the number of licenses that your company has purchased.

Important

Overwrites gis_interactionservice-block-size option value for the first request.

gis_interactionservice-maxblock-size

- Default Value: 0 (no maximum)
- Valid Values: Any positive value.
- **Description:** Specifies the maximum number of interaction corporate licenses for connections to FlexLM server. The valid value must not exceed the number of licenses that your company has purchased.

gis_statSERVICE-block-size

- Default Value: 1
- Valid Values: Any positive value.
- **Description:** Specifies the number of statistic corporate licenses for connections to FlexLM server. The valid value must not exceed the number of licenses that your company has purchased.

gis_statSERVICE-firstblock-size

- Default Value: 0 (no specific first request)
- Valid Values: Any positive value.
- **Description:** Specifies the number of statistic corporate licenses for the first request to FlexLM server. The valid value must not exceed the number of licenses that your company has purchased.

Important

Overwrites gis_statSERVICE-block-size option value for the first request.

gis_statSERVICE-maxblock-size

- Default Value: 0 (no maximum)
- Valid Values: Any positive value.
- **Description:** Specifies the maximum number of statistic corporate licenses for the requests to FlexLM server. The valid value must not exceed the number of licenses that your company has purchased.

gis_configurationSERVICE-block-size

- Default Value: 1
- Valid Values: Any positive value.

- **Description:** Specifies the maximum number of configuration service corporate licenses for connections to FlexLM server. The valid value must not exceed the number of licenses that your company has purchased.

gis_configurationservice-firstblock-size

- Default Value: 0 (no specific first request)
- Valid Values: Any positive value.
- **Description:** Specifies the maximum number of configuration service corporate licenses for connections to FlexLM server. The valid value must not exceed the number of licenses that your company has purchased.

Important

Overwrites gis_configurationservice-block-size option value for the first request.

gis_configurationservice-maxblock-size

- Default Value: 0 (no maximum)
- Valid Values: Any positive value.
- **Description:** Specifies the maximum number of configuration service corporate licenses for connections to FlexLM server. The valid value must not exceed the number of licenses that your company has purchased.

license-file

- Default Value: license.dat
- Valid Values: *port@hostname1*, *port@hostname2*
- **Description:** Addresses of FlexLM license servers.

log Section

Important

For GIS:SOAP deployments, the options you specify in the GIS Application object control logs for both GIS and its related Services (that is, Agent Interaction Services and Open Media Interaction Services) in the same output. The log options set in the

connected Services Application objects do not apply.

all

- Default Value: network, ../logs/all.log, stdout
- Valid Values: network, ../logs/all.log, all.log, stdout
- **Description:** Sets output options for the all log level. The values are each optional, and can be combined:
 - network specifies that these logs will be output to MessageServer if GIS is connected to it.
 - ../logs/all.log specifies an output file for this log level, for a GIS:SOAP connector.
 - stdout specifies that this log level will be output to a console if the console is available.

Warning

If you have deployed a GIS:SOAP connector as a Web Module, replace the default relative path to the log file (../logs/all.log) with a fully qualified path/file name, of the form: <logpath>/<logfilename>.log

Buffering

- Default Value: true
- Valid Values: true, false
- Changes Take Effect: immediately
- **Description:** If set to true, enables operating system file buffering. (This option only applies to stderr and stdout output.)
File buffering improves output performance, but can result in the loss of buffered logs in the case of a server crash.

debug

- Default Value: [none]
- Valid Values: network, ../logs/all.log, all.log, stdout
- **Description:** Sets output options for the debug log level. The values are each optional, and can be combined:
 - network specifies that these logs will be output to MessageServer if GIS is connected to it.
 - ../logs/all.log specifies an output file for this log level, for a GIS:SOAP connector.

- `stdout` specifies that this log level will be output to a console if the console is available.

Warning

If you have deployed a GIS:SOAP connector as a Web Module, replace the default relative path to the log file (`../logs/all.log`) with a fully qualified path/file name, of the form: `<logpath>/<logfilename>.log`

expire

- Default Value: `false`
- Valid Values:
 - `false`: No expiration; all generated segments are stored.
 - `<number>`, or `<number> file`: Sets the maximum number of log files to store. Specify a number from 1-100.
 - `<number> day`: Sets the maximum number of days before log files are deleted. Specify a number from 1-100.
- Changes Take Effect: immediately.
- **Description:** Determines whether log files will be deleted, and if so, the deletion interval (set as a maximum number of log files or retention days).

MessageFile

- Default Value: [As specified by a particular application.]
- Valid Values: `<string> .lms` (message file name)
- Changes Take Effect: Immediately, in the case when an application cannot find its
 - `.lms`file at startup.
- **Description:** Specifies the file name for application-specific log events. The name should be valid for the operating system on which the application is running. The option value can also contain the absolute path to the application-specific
 - `.lms`

. Otherwise, an application looks for the file in its working directory.

Warning

An application that cannot find its

- `.lms`

file at startup cannot generate application-specific log events or send them to Message Server.

segment

- Default Value: `false`
- Valid Values:
 - `false`: No segmentation allowed.
 - '`<number>`' or '`<number>`' KB: Sets maximum segment size, in kilobytes. (The minimum segment size is 100 KB.)
 - `<number>` MB: Sets maximum segment size, in megabytes.
 - `<number>` hr: Sets maximum segment size, in hours. (The minimum segment size is 1 hour.)
- Changes Take Effect: Immediately.
- **Description:** Specifies whether log file has a segmentation limit, and if so, the segment maximum (measured in size or elapsed time). If the current log segment exceeds the size set by this option, the current file is closed and a new one is created.

standard

- Default Value: `[none]`
- Valid Values: `network`, `../logs/all.log`, `all.log`, `stdout`
- **Description:** Sets output options for the standard log level. The values are each optional, and can be combined:
 - `network` specifies that these logs will be output to MessageServer if GIS is connected to it.
 - `../logs/all.log` specifies an output file for this log level, for a GIS:SOAP connector.
 - `stdout` specifies that this log level will be output to a console if the console is available.

Warning

If you have deployed a GIS:SOAP connector as a Web Module, replace the default relative path to the log file (`../logs/all.log`) with a fully qualified path/file name, of the form: `<logpath>/<logfilename>.log`

trace

- Default Value: [none]
- Valid Values: network, ../logs/all.log, all.log, stdout
- **Description:** Sets output options for the trace log level. The values are each optional, and can be combined:
 - network specifies that these logs will be output to MessageServer if GIS is connected to it.
 - ../logs/all.log specifies an output file for this log level, for a GIS:SOAP connector.
 - stdout specifies that this log level will be output to a console if the console is available.

Warning

If you have deployed a GIS:SOAP connector as a Web Module, replace the default relative path to the log file (../logs/all.log) with a fully qualified path/file name, of the form: <logpath>/<logfilename>.log

verbose

- Default Value: trace
- Valid Values: all, debug, standard, trace
- **Description:** Sets the target log's level.

multimedia Section

logout-on-last-media

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

If true, the place is automatically logged out from the Interaction Server when the last remaining media is logged out. If false, the agent can remain logged in the place with no media from an Interaction Server point of view.

session-services Section

sessionTimeout

- Default Value: 3600
- Valid Values: 0 to 2,147,483,647
- Changes Take Effect: Immediately.

This sets a timeout, in seconds, for incoming POST HTTP requests. After expiration, the agent is logged out. A value of 0 (zero) disables the timeout; a value cannot be negative.

stat-services Section

restriction_time

- Default Value: 30 seconds
- Valid Values: 3 seconds or greater
- **Description:** This parameter specifies how long the client must wait between retrieveSubscribedStatistic requests.

error_check

- Default Value: true
- Valid Values: true or false
- **Description:** This option establishes whether GIS uses error checking during statistics subscriptions. To disable error checking, set this value to false.

Important

If you are experiencing exceptionally long subscription times, set this value to false. In that case, subscriptions are allowed even if the parameters you specified are incorrect. You will receive an error message when you try to retrieve statistics with invalid parameters. You must then unsubscribe to the invalid statistic.

scopeStatEvents (optional)

- Default Value: 15
- Valid Values:
- **Description:** Use this option to set the maximum number of statistic events that a client application can retrieve for one statistic.