Genesys Integration Server 7.6

Deployment Guide

About Genesys

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

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

Notice

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

Your Responsibility for Your System's Security

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

Trademarks

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

The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

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

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys Licensing Guide.***

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 76gis_dep_07-2013_v7.6.502.00

Table of Contents

List of Procedures

Preface

Welcome to the *Genesys Integration Server 7.6 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

This document is valid only for the 7.6 release(s) of this product.

Note: For versions of this document created for other releases of this product, please visit the Genesys Technical Support website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

This preface provides an overview of this document, identifies the primary audience, introduces document conventions, and lists related reference information:

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.

Usage Guidelines

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

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

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

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

1. Possession of interface documentation does not imply a right to use by a third party. Genesys conditions for use, as outlined below or in the *Genesys Developer Program Guide,* must be met.

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

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

The Genesys developer materials outlined in this document are not intended to be used for the creation of any product with functionality comparable to any

Genesys products, including products similar or substantially similar to Genesys's current general-availability, beta, and announced products.

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

Chapter Summaries

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

- Chapter 1, "Introducing GIS," on page 17, presents an overview of GIS functionality and architecture, plus deployment-planning information.
- Chapter 2, "Installing and Uninstalling GIS," on page 39, provides step-by-step instructions for installing, uninstalling, and deploying GIS.
- Chapter 3, "Customizing the GIS Environment," on page 59, provides options for reconfiguring GIS after installation.
- Chapter 4, "High-Availability Deployments of GIS," on page 73, provides details on how to deploy GIS in high-availability (HA) mode, and describes the load balancing capabilities of those deployments.
- Chapter 5, "Starting and Testing GIS," on page 91, provides step-by-step instructions for starting and stopping GIS and testing your installation.
- Appendix A, "Configuration Options," on page 103, configuration information for the options tab.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

76gis_dep_02-2008_v7.6.001.00

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

Type Styles

Italic

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

Examples:

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

Monospace Font

A monospace font, which looks like `teletype or typewriter text`, is used for all programming identifiers and GUI elements.

This convention includes the *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages; the values of options; logical arguments and command syntax; and code samples.

Examples:

- Select the `Show variables on screen` check box.
- Click the `Summation` button.
- In the `Properties` dialog box, enter the value for the host server in your environment.
- In the `Operand` text box, enter your formula.
- Click `OK` to exit the `Properties` dialog box.
- The following table presents the complete set of error messages T-Server distributes in `EventError` events.
- If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls.

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

Example:

- Enter `exit` on the command line.

Screen Captures Used in This Document

Screen captures from the product GUI (graphical user interface), as used in this document, may sometimes contain a minor spelling, capitalization, or grammatical error. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from

installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Square Brackets

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

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

Angle Brackets

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

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

Related Resources

Consult these additional resources as necessary:

- *Statistics SDK* 7.6 *Web Services API Reference* for information specific to developing a client application that interfaces with the Statistics Service in GIS.
- *Statistics SDK* 7.6 *Web Services Developer's Guide* for instructions on installing and running the Statistics SDK code examples, along with the example code and explanatory comments.
- *Configuration SDK* 7.6 *Web Services API Reference* for information specific to developing a client application that interfaces with the Configuration Service in GIS.
- *Configuration SDK* 7.6 *Web Services Developer's Guide* for instructions on installing and running the Configuration SDK code examples, along with the example code and explanatory comments.
- *Interaction SDK* 7.6 *Java Deployment Guide* for detailed compatibility requirements for the Agent Interaction services and Open Media Interaction services.
- *Agent Interaction SDK* 7.6 *Web Services API Reference for the Java Proxy,* which provides API details about the connection proxy that Genesys provides for the protocol.

- *Agent Interaction SDK* 7.6 *Web Services API Reference for the .NET Proxy,* which provides API details about the connection proxy that Genesys provides for its proprietary GSAP protocol.
- *Framework* 7.6 *Configuration Manager Help* for assistance in using Configuration Manager.
- *Framework* 7.6 *Solution Control Interface Help* for specific instructions on using Solution Control Interface to start, stop, and monitor GIS.
- *Genesys Desktop* 7.6 *Application Blocks for .NET.* These offer models of useful applications and modules that you can build using the Genesys 7.6 SDKs. Particularly relevant are the *Genesys Desktop* 7.6 *Connection Application Block* and the *Genesys Desktop* 7.6 *Statistics Service Application Block.*
- *Multimedia* 7.6 *Deployment Guide* for deployment procedures for the Genesys Multimedia solution and its components.
- The *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.
- The *Genesys Migration Guide,* also on the Genesys Documentation Library DVD, which provides a documented migration strategy from Genesys product releases 5.1 and later to all Genesys 7.x releases. Contact Genesys Technical Support for additional information.
- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at http://genesyslab.com/support.

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

- *Genesys Supported Operating Environment Reference Guide*
- *Genesys Supported Media Interfaces Reference Manual*

Genesys product documentation is available on the:

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

Making Comments on This Document

If you especially like or dislike anything about this document, please feel free to e-mail your comments to Techpubs.webadmin@genesyslab.com.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document.

Please limit your comments to the information in this document only and to the way in which the information is presented. Speak to Genesys Technical Support if you have suggestions about the product itself.

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

Contacting Genesys Technical Support

If you have purchased support directly from Genesys, please contact Genesys Technical Support.

Before contacting technical support, please refer to the *Genesys Care Program Guide* for complete contact information and procedures.

Chapter

1 Introducing GIS

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

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.

Note: 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 2 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).

Depending on your client applications' requirements, any given instance of GIS can be configured and installed to use the Genesys proprietary GSAP protocol instead of SOAP.

Note: The GSAP protocol offers better performance than SOAP, but GSAP requires a special port. SOAP runs on Hypertext Transfer Protocol (HTTP) ports.

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" on page 23.

Warning! When deploying GIS in GSAP stand-alone mode, ensure that you first install Java Standard Edition—*not* Java Enterprise Edition—on the host machine. (This is a separate matter from GIS's J2EE compliance). GIS requires certain JMX components that are not included in the Enterprise Edition. 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 a *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, found on the Genesys Developer Documentation Library DVD, 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.

> **Note:** 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.

Warning! Route Point and Route Service functionality are restricted within the Agent Interaction services, as deployed on GIS.

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

Figure 1 on page 22 shows the connections between GIS, the Genesys Framework, and your client application.

Notes: All the APIs and services shown in Figure 1 on page 22 are available to GIS:SOAP installations. Those labeled with an asterisk (*) are available only to GIS:SOAP installations—not to GIS:GSAP 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.

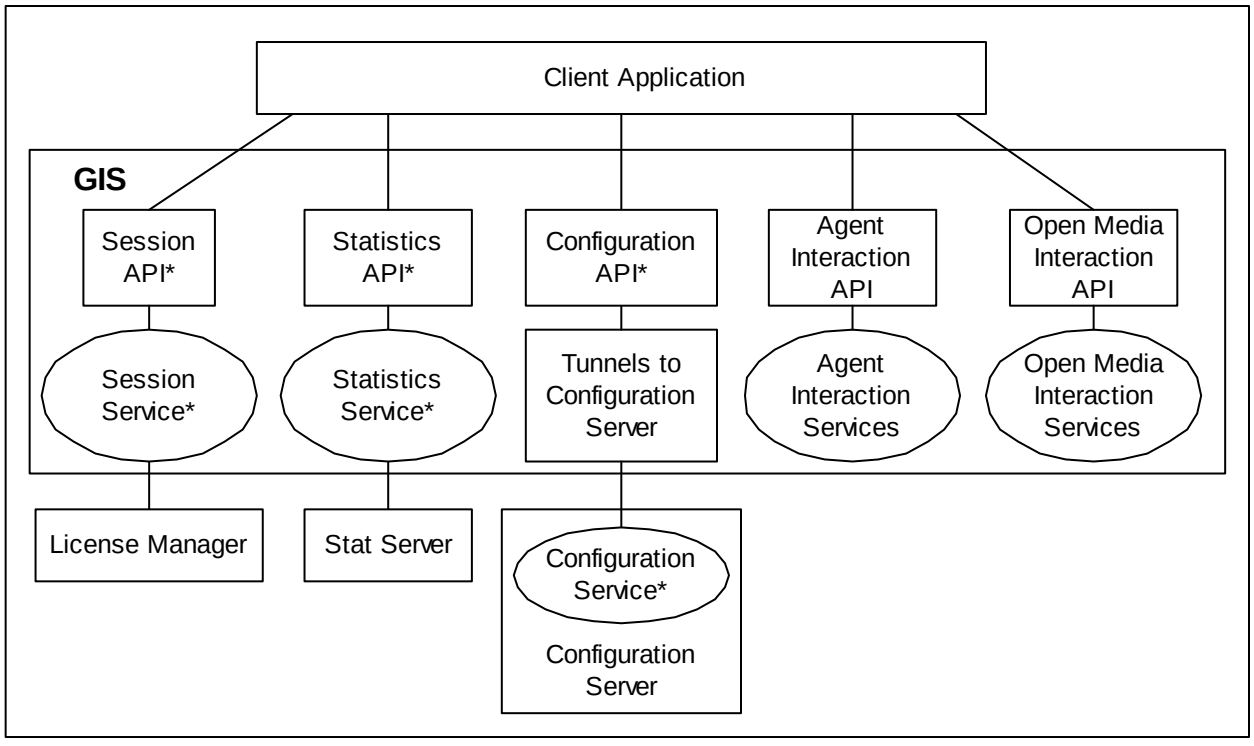

Figure 1: GIS Architecture—APIs and Services marked with an asterisk (*) are only available to SOAP installations.

GIS—Built with Flexible Technologies

GIS client applications communicate with GIS either using SOAP over HTTP, or using the proprietary Genesys Services Access Protocol (GSAP). 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 both the SOAP or GSAP protocols). These proxies/libraries incorporate reconnect and defense mechanisms, as well as performance enhancements.

If you choose to deploy 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 3.0, 3.5, or 4.0 (`http://msdn.microsoft.com/netframework/`)
- Apache AXIS toolkit, version 1.3 (`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 512 MB RAM.
- Have a CPU speed above 1.5 GHz, with 1.7 GHz as a minimum for Windows platforms.

> Note: 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, especially for SOAP deployments.
- Run the Java Runtime environment (JRE) or Java SDK, version 1.4 or 1.5:
  - Windows and Solaris users should download the necessary software from the Sun Java site at `http://java.sun.com`.
  - AIX users should download the necessary software from `http://www-106.ibm.com/developerworks/java/jdk/aix/`.
  - HP-UX users should download the necessary software from `http://www.hp.com/products1/unix/java/`.
  - Tru64 UNIX users should download the necessary software from `http://h18012.www1.hp.com/java/alpha/`.

> Warnings! 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 1.5.0_10 or higher.
> - When deploying GIS using the GSAP option, first install Java Standard Edition—*not* Java Enterprise Edition—on the host machine. GIS requires certain JMX components that are not included in the Enterprise Edition. 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 5.5 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 and 45 MB for GSAP deployments, with slight variations depending upon operating system.
- If you implement high availability using GIS:GSAP, you will need a JDBC-compatible RDBMS (relational database management system), in which you will create a recovery database. Supported databases are Oracle 9.2i (or higher), and Microsoft SQL Server 2000 or SQL Server 2005.
- 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.

> **Note:** 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

> Note: 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

> Note: 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.

> Note: 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, Figure 2.

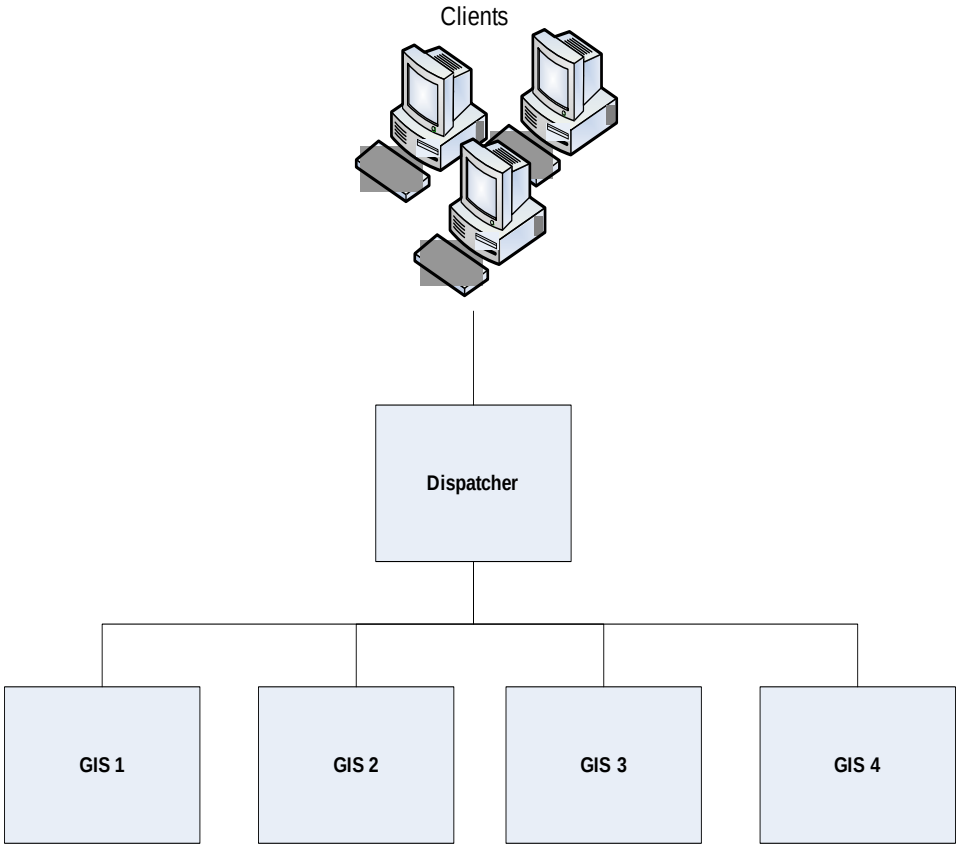

Figure 2: GIS:SOAP 7.6 HA deployment using a GIS cluster.

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.

Note: For more information about this deployment (See "Deployment for High Availability with Load Balancing" on page 80).

Example from a Load Balanced and Highly Available Environment

Figure 3 shows a multi-site, multi-host, multi-cluster, high availability environment that supports two-thousand (2000) agents.

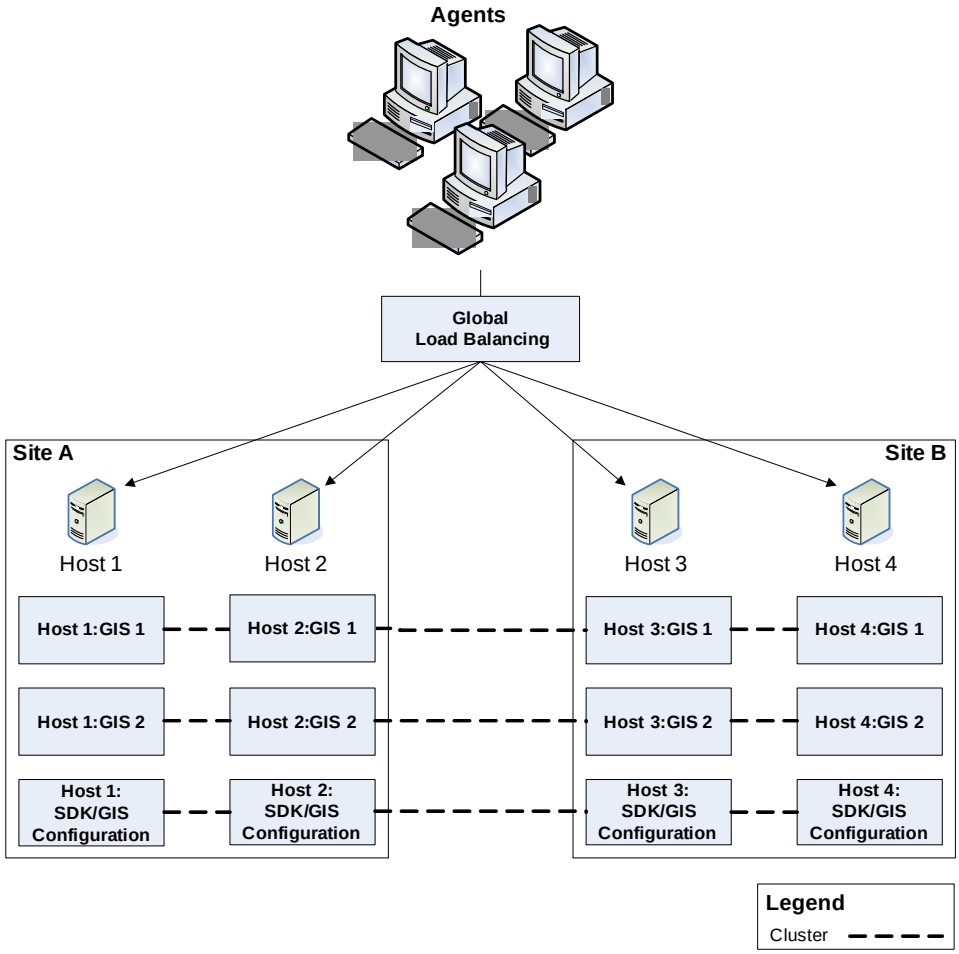

Figure 3: Multi-site GIS cluster with HA

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 (GBL)

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.

Note: 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).

Notes: 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 or GIS:GSAP), page 94, which contains instructions on how to configure license information.
- Specifying Licensing at Startup (GIS:SOAP only), page 95, which explains how to specify license information when starting GIS.
- 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.

New In This Release

This section summarizes what has changed between Genesys Integration Server 7.6 and its predecessor products. It identifies new options and compatibility issues.

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.

GIS:GSAP High Availability Support

GIS:GSAP 7.6 supports high-availability. This functionality provides failover to one or more backup GIS:GSAP servers.

Note: GIS high-availability deployment is intended to support failover scenarios, not to support manual switchover among servers.

For details, see Chapter 4, “High-Availability Deployments of GIS,” on page 73.

Tasks and their Related Procedures

Table 1 on page 33 summarizes the tasks and their related procedures addressed in this guide.

Table 1: Tasks and Related Procedures

Objective	Related Procedures and Actions
To install GIS as a GSAP server, a SOAP stand-alone server, or a SOAP web module on Solaris, HP-UX, Linux, AIX, or Tru64 UNIX platforms.	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. You must create three application objects, one for Agent Interaction Service, one for Open Media Service, and one for GIS. For information regarding importing templates and creating application objects, see the *Genesys Framework Deployment Guide*. The following configuration procedures must be completed for each of your application objects: • 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). For detailed information, refer to the *Genesys Framework Deployment Guide*. • Configure the Options Tab. For detailed information on the options available for your application object, refer to: • “Agent Interaction Service” on page 103. • “Open Media Interaction Service” on page 110. • “Genesys Integration Server” on page 113. Follow the steps in Installing GIS on a UNIX-Based Platform, page 40.

Table 1: Tasks and Related Procedures (Continued)

Objective	Related Procedures and Actions
To install GIS as a GSAP server, a SOAP stand-alone server, or a SOAP web module on a Windows platform.	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. You must create three application objects, one for Agent Interaction Service, one for Open Media Service, and one for GIS. For information regarding importing templates and creating application objects, see the *Genesys Framework Deployment Guide*. The following configuration procedures must be completed for each of your application objects: • 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). For detailed information, refer to the *Genesys Framework Deployment Guide*. • Configure the Options Tab. For detailed information on the options available for your application object, refer to: • "Agent Interaction Service" on page 103. • "Open Media Interaction Service" on page 110. • "Genesys Integration Server" on page 113. Follow the steps in Installing GIS on a Windows Platform, page 43.

Table 1: Tasks and Related Procedures (Continued)

Objective	Related Procedures and Actions
To run GIS in the BEA WebLogic application server.	• Deploy GIS in the WebLogic application server. Deploying GIS in WebLogic, page 45. • Verify the successful deployment of GIS in WebLogic.Inspecting the WebLogic Directory Tree, page 52. • Confirm that the GIS server has started successfully in WebLogic. Checking the GIS Log Files, page 53.
To run GIS in WebSphere application server.	• Deploy GIS in the WebSphere. Deploying GIS in WebSphere, page 46. • Verify the successful deployment of GIS in WebSphere. Inspecting the WebSphere Directory Tree, page 50. • Confirm that the GIS server has started successfully in WebLogic. Checking the GIS Log Files, page 53.
To verify the successful installation of GIS:SOAP in Tomcat (stand-alone).	Inspecting the Stand-alone Directory Tree, page 49.
To verify the successful installation of GIS as a GSAP sever.	Verifying Installation of GIS as a GSAP Server, page 48.
To uninstall GIS.	• From a Windows platform for GSAP or stand-alone server installations, Uninstalling GIS as a GSAP or SOAP Stand-alone Server from a Windows Platform, page 54. • From a UNXI-based platform for GSAP or stand-alone server installations, Uninstalling GIS as a GSAP or SOAP Stand-alone Server from a UNIX-Based Platform, page 55. • From a Web application server, Uninstalling GIS from a WebSphere Application Server, page 55.

Table 1: Tasks and Related Procedures (Continued)

Objective	Related Procedures and Actions
To deploy GIS with the High Availability option.	• To create a standard node cluster *without* load balancing follow the procedures listed in "Deployment for High Availability (No Load Balancing)" on page 74. • To create a standard node cluster *with* load balancing follow the procedures listed in "Deployment for High Availability with Load Balancing" on page 80. • To create a Primary/Backup GIS pair follow the procedures listed in "Alternative Deployment for High Availability" on page 84.
To customize your applications configuration.	You can customize your application in various ways using the following procedures: "Windows Service Setting" on page 59. "LCA Configuration" on page 60. "Customize the WSDL Files" on page 68. "SDK Server Connections for GIS SOAP" on page 62. Configuring a SOAP Port for a Master Configuration Server, page 63. Configuring a SOAP Port for a CS Proxy, page 64. Customizing the Notification Mechanism for GIS:SOAP, page 65. Displaying the WSDL Files for GIS:SOAP, page 68.

Table 1: Tasks and Related Procedures (Continued)

Objective	Related Procedures and Actions
To start/test GIS.	Choosing Your Start Script, page 91 Editing the Start Script, page 92. Starting and Testing GIS, page 93. For GIS:SOAP: • Specifying Licensing at Startup (GIS:SOAP only), page 95. • Starting on UNIX-based Platforms (GIS:SOAP only), page 96 • Verify the Status of GIS (GIS:SOAP Only), page 96. For Windows Platforms: • Starting GIS on Windows, page 98. For Web Server Applications: • Starting GIS on a Web Application Server, page 98. Using Management Layer: Using Management Layer to Start and Stop GIS, page 101.
To stop GIS.	For Windows Platforms, follow Stopping GIS on Windows, page 100. For UNIX-based Platforms, follow Stopping GIS on UNIX-based Platforms, page 99. For Web Application Servers, follow Stopping on a Web Application Server, page 100. For Management Layer, follow Using Management Layer to Start and Stop GIS, page 101.

Chapter

1 Installing and Uninstalling GIS

This chapter describes how to install, uninstall, and deploy the Genesys Integration Server (GIS) `Application` object. It contains the following sections:

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 GSAP server, a SOAP stand-alone server, or a SOAP web module on Solaris, HP-UX, Linux, AIX, or Tru64 UNIX platforms:
  - "Installing GIS on a UNIX-Based Platform" on page 40.
- To install GIS as a GSAP server, a SOAP stand-alone server, or a SOAP web module on Windows platforms:
  - "Installing GIS on a Windows Platform" on page 43.
- To deploy GIS into an IBM WebSphere application server:
  - "Deploying GIS in WebSphere" on page 46.
- To deploy GIS into the BEA WebLogic application server on the Solaris platform:
  - "Deploying GIS in WebLogic" on page 45.

Procedure:
Installing GIS on a UNIX-Based Platform

Purpose: To install GIS as a GSAP server, a SOAP stand-alone server, or a SOAP web module on Solaris, HP-UX, Linux, AIX, or Tru64 UNIX platforms.

Prerequisites

- Before selecting a GIS:GSAP Server installation:
  - Configure at least one GIS:GSAP application in the Configuration Layer; otherwise, the installer will abort when it cannot find an Application object. Refer to the *Genesys Framework Deployment Guide* for information on how to create and configure an application object.
  - Certain GIS services require a GIS:SOAP installation, see Figure 1 on page 22 and check whether your client application needs one of these services.
- 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" on page 57.
  - 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 Deployment Guide* for information on how to create an application object.
  - Install and start your application server.

Note: 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 of procedure

1. Open the appropriate folder for your platform on the installation CD. The folders are:
   - `GIS/aix/`

- GIS/linux/
- GIS/hp-ux/
- GIS/solaris/
- GIS/tru64UNIX/

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 GSAP Server, select: `2) GSAP (Genesys Service Access Protocol).`
   - 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 GSAP Server, the installation type prompt will not be displayed.
   - 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.

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

11. When prompted for the `shutdown port`:
    - for GIS as a GSAP Server, the `shutdown port` prompt will not be displayed.
    - 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.
12. When prompted for the `destination location` for this installation:
    - for GIS as a GSAP Server, specify the path.
    - 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 GSAP or 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.

> Note: GIS does not support installation paths that contain spaces.

13. When prompted for the target directory for the Web Application Archive:
    - for GIS as a GSAP Server, the target directory for the Web Application Archive prompt will not be displayed.
    - 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.
14. Follow the script's remaining prompts to complete the installation.

End of procedure

Next Steps

- To verify the installation of GIS, you can select from one of the following procedures:
  - "Verifying Installation of GIS as a GSAP Server" on page 48.
  - "Inspecting the Stand-alone Directory Tree" on page 49.
- To deploy your GIS:SOAP Web Module, choose one of the following procedures:
  - "Deploying GIS in WebSphere" on page 46.
  - "Deploying GIS in WebLogic" on page 45.

Procedure:
Installing GIS on a Windows Platform

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

Prerequisites

- Before selecting a GIS:GSAP Server installation:
  - Configure at least one GIS:GSAP application in the Configuration Layer; otherwise, the installer will abort when it cannot find an Application object. Refer to the *Genesys Framework Deployment Guide* for information on how to create and configure an application object.
  - Certain GIS services require a GIS:SOAP installation, see Figure 1 on page 22 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" on page 57.
  - 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.

Note: 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 of procedure

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 GSAP or SOAP Stand-alone Server from a Windows Platform" on page 54 for details).

5. On the `Select installation type` page, select:
   - for GIS as a GSAP Server—`Deploy GSAP (binary protocol)`
   - 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`.

   GIS does not support installation paths that contain spaces.

9. On the `Genesys Integration Server Parameters` page, confirm or override:
   - for GIS as a GSAP Server—the default `Shutdown Port` for GIS:GSAP
   - 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" on page 59.)

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

End of procedure

Next Steps

- To verify the installation of GIS, you can select from one of the following procedures:
  - "Verifying Installation of GIS as a GSAP Server" on page 48.
  - "Inspecting the Stand-alone Directory Tree" on page 49.
- To deploy your GIS:SOAP Web Module, choose one of the following procedures:
  - "Deploying GIS in WebSphere" on page 46.
  - "Deploying GIS in WebLogic" on page 45.

Procedure: 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" on page 40. as a guide.

Warnings! 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 of procedure

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

   For UNIX-based installations, this file is located in the directory specified in Step 13 of "Installing GIS on a UNIX-Based Platform" on page 40.

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

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

3. Start the WebLogic Server Administration Console for the domain in which you will be working.
4. 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.

5. 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.

6. Click `Target Application`.

7. 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.
8. Enter `gis` in the `Name` field as the name for the Web Application.

> Note: 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.)

9. 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`.
10. Point your browser to the following URL and verify that the GIS connector is starting:

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

End of procedure

Next Steps

- If you want to confirm that GIS has started successfully "Checking the GIS Log Files" on page 53.
- To confirm that the installation of GIS in WebLogic was correct and complete, "Inspecting the WebLogic Directory Tree" on page 52.
- To customize your GIS environment, Chapter 3, "Customizing the GIS Environment," on page 59.
- To add High Availability and Load Balancing to your GIS environment, Chapter 4, "High-Availability Deployments of GIS," on page 73.
- To start and test GIS, Chapter 5, "Starting and Testing GIS," on page 91.

Procedure: Deploying GIS in WebSphere

Purpose: To deploy GIS is WebSphere regardless on platform type.

Prerequisites

- You must have and GIS installed. Use the procedures provide to install GIS according to your platform requirements:

- “Installing GIS on a UNIX-Based Platform” on page 40.
- “Installing GIS on a Windows Platform” on page 43.

Start of procedure

1. Open the WebSphere Administration Console, as follows:
   - For WebSphere Express 5.x, this console is configured to use port `7090` by default. In this case, the URL is:
     `http://<WebSphere_host>:7090/admin/`
   - For WebSphere Base 5.x, this console is configured to use port `9090` by default. In this case, the URL is:
     `http://<WebSphere_host>:9090/admin/`
   - 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.

Note: 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/` subdirectory.

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 of procedure

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)” on page 54.
- If you are not deploying in WebSphere 5.0 then you must verify your installing by following the “Inspecting the WebSphere Directory Tree” on page 50 procedure.
- To customize your GIS environment, Chapter 3, “Customizing the GIS Environment,” on page 59.
- To add High Availability and Load Balancing to your GIS environment, Chapter 4, “High-Availability Deployments of GIS,” on page 73.
- To start and test GIS, Chapter 5, “Starting and Testing GIS,” on page 91.

Procedure: Verifying Installation of GIS as a GSAP Server

Purpose: To confirm that the installation of GIS as a GSAP Server was correct and complete.

Prerequisites

- You must have installed GIS as a GSAP Server on either your UXIN-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” on page 40.
  - For Windows platforms: “Installing GIS on a Windows Platform” on page 43.

Start of procedure

1. After completing installation, inspect your system's directory tree to make sure that the files have been installed in the location you intended.

End of procedure

Next Steps

- To customize your GIS environment, Chapter 3, "Customizing the GIS Environment," on page 59.
- To add High Availability and Load Balancing to your GIS environment, Chapter 4, "High-Availability Deployments of GIS," on page 73.
- To start and test GIS, Chapter 5, "Starting and Testing GIS," on page 91.

Procedure: 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 UXIN-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" on page 40.
  - For Windows platforms: "Installing GIS on a Windows Platform" on page 43.

Start of procedure

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 of procedure

Next Steps

- To customize your GIS environment, Chapter 3, “Customizing the GIS Environment,” on page 59.
- To add High Availability and Load Balancing to your GIS environment, Chapter 4, “High-Availability Deployments of GIS,” on page 73.
- To start and test GIS, Chapter 5, “Starting and Testing GIS,” on page 91.

Procedure: 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" on page 46.

Start of procedure

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 of procedure

Next Steps

- To customize your GIS environment, Chapter 3, “Customizing the GIS Environment,” on page 59.
- To add High Availability and Load Balancing to your GIS environment, Chapter 4, “High-Availability Deployments of GIS,” on page 73.
- To start and test GIS, Chapter 5, “Starting and Testing GIS,” on page 91.

Procedure: 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” on page 45.

Start of procedure

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 of procedure

Next Steps

- If you want to confirm that GIS has started successfully “Checking the GIS Log Files” on page 53.
- To customize your GIS environment, Chapter 3, “Customizing the GIS Environment,” on page 59.
- To add High Availability and Load Balancing to your GIS environment, Chapter 4, “High-Availability Deployments of GIS,” on page 73.
- To start and test GIS, Chapter 5, “Starting and Testing GIS,” on page 91.

Procedure: 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” on page 45.

Start of procedure

1. In the WebLogic Server Console’s navigation tree, select the server that you created in Step 8 on page 46.
2. Select the `Control` tab.
3. Select `Remote Start Output`.
4. Select `View Server Output`.

End of procedure

Next Steps

- To customize your GIS environment, Chapter 3, “Customizing the GIS Environment,” on page 59.
- To add High Availability and Load Balancing to your GIS environment, Chapter 4, “High-Availability Deployments of GIS,” on page 73.
- To start and test GIS, Chapter 5, “Starting and Testing GIS,” on page 91.

Procedure:
Replacing the JavaMail Archive (for WebSphere 5.0)

Purpose: To replace WebSphere's default JavaMail archive (version 1.2) with version 1.3 which is supplied with the GIS application.

Start of procedure

1. Locate the default JavaMail archive is located at: *<WebSphere_home>*/java/jre/lib/ext/mail.jar.
2. Stop WebSphere Application Server.
3. Make a backup copy of the mail.jar archive identified above.
4. Overwrite the mail.jar archive identified above with the version installed by GIS at: <WebSphere_home>/installedApps/<node_name>/gis.war/<application_name>/WEB-INF/lib/mail.jar.
5. Restart WebSphere Application Server.

End of procedure

Next Steps

- To verify correct installation, "Inspecting the WebSphere Directory Tree" on page 50.
- To customize your GIS environment, Chapter 3, "Customizing the GIS Environment," on page 59.
- To add High Availability and Load Balancing to your GIS environment, Chapter 4, "High-Availability Deployments of GIS," on page 73.
- To start and test GIS, Chapter 5, "Starting and Testing GIS," on page 91.

Procedure:
Uninstalling GIS as a GSAP or SOAP Stand-alone Server from a Windows Platform

Purpose: To uninstalling GIS as a GSAP or SOAP Stand-alone Server from a Windows Platform.

Prerequisites

- You must have already installed GIS as a GSAP or SOAP Stand-alone Server using one of the following procedure, "Installing GIS on a Windows Platform" on page 43.

Start of procedure

1. Stop GIS. (See "Stopping GIS" on page 99 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 of procedure

Procedure: Uninstalling GIS as a GSAP or SOAP Stand-alone Server from a UNIX-Based Platform

Purpose: To uninstalling GIS as a GSAP or SOAP Stand-alone Server from a UNIX-based Platform.

Prerequisites

- You must have already installed GIS as a GSAP or SOAP Stand-alone Server using one of the following procedure:
  - "Installing GIS on a UNIX-Based Platform" on page 40.

Start of procedure

1. Stop GIS. (See "Stopping GIS" on page 99 for details.)
2. Open the GIS home directory and manually remove all files.

End of procedure

Procedure: 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" on page 46.

Start of procedure

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 of procedure

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.

Procedure: 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" on page 45.

Start of procedure

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 of procedure

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.

Procedure: 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.

Note: 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 of procedure

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.
   - If you have deployed GIS on a 64-bit version of Windows 2003 Server, point to a 32-bit version of Java Virtual Machine (JVM) 1.5.
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 of procedure

Next Steps

- Create and configure an instance of GIS. See the *Genesys Framework Deployment Guide* for details.

Chapter

3 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. This chapter contains the following sections:

Windows Service Setting

If you deploy GIS as a SOAP or GSAP stand-alone server, 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 7.x Deployment Guide*.

Note: 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.

Load Balancing Under GSAP

If you have deployed multiple GIS:GSAP connectors in a high-availability cluster, your primary GIS:GSAP instance functions as the load-balancing dispatcher. It collects load information from the other GIS:GSAP instances in its cluster, and uses this information to balance the load of client requests across the cluster.

You can fine-tune this load-balancing behavior by configuring your GIS `Application` options, as outlined in the "[GSAP.LoadBalancing] Section". In general, load balancing proceeds as follows:

1. A client application sends a connection request to your primary GIS:GSAP instance (the default connector).
2. This primary GIS:GSAP connector polls the other GIS:GSAP connectors in its cluster for load information.
3. The primary connector calculates which connector in the cluster has the lowest load, as a percentage of its configured `lb-load-threshold` maximum value (see "GSAP Load Balancing Configuration" on page 61).

   Note: This calculation excludes any GIS:GSAP connector that is currently down.

- The primary connector allocates the client request to this lowest-load connector.

GSAP Load Balancing Configuration

When configuring the GSAP load balancing options, you should remember that the estimated load can be calculated using the following algorithm:

(number of connections / `lb-max-connections-per-server`) * 100

The following sample shows how to configure an equally-balanced cluster of four GIS instances (two per machine):

```
[GSAP.LoadBalancing]
lb-clientsocket-enabletcpnodelay=true
lb-clientsocket-sotimeout=5000
lb-cluster-servers=gisserver1:1201, gisserver1:1202,
  gisserver2:1201, gisserver2:1202
lb-connector-exchange-interval=5000
lb-connector-process-interval=30
lb-load-delta-threshold=20
lb-load-threshold=5
lb-max-connections-per-server=1000
lb-max-transfer-count=3
lb-port=1201
lb-process-interval=0
```

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

Procedure: Tuning Heap Size Values

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

Start of procedure

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 and at the root of the installation for GIS:GSAP.
2. Search for and set the `xmx/xms` parameters according to your environment.

End of procedure

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 Table 2.

Table 2: Configuration SDK Server Connections

Configuration Server Version	Application Objects and Connections
7.6, 7.5 or 7.2	• 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 CS_Proxy 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	• 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 CS_Proxy 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.
6.5	• 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.

Note: See your Framework documentation for additional details about configuring Configuration Server options.

Procedure: 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 of procedure

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 of procedure

Next Steps

- Customize your GIS application further by:
  - "Windows Service Setting" on page 59.
  - "LCA Configuration" on page 60.
  - "SDK Server Connections for GIS SOAP" on page 62.
  - "Configure the Configuration Server SOAP Port" on page 63.
  - "Customize the Notification Mechanism" on page 65.
  - "Customize the WSDL Files" on page 68.

Procedure: 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 of procedure

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`.

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

End of procedure

Next Steps

- Customize your GIS application further by:
  - "Windows Service Setting" on page 59.
  - "LCA Configuration" on page 60.
  - "SDK Server Connections for GIS SOAP" on page 62.
  - "Configure the Configuration Server SOAP Port" on page 63.
  - "Customize the Notification Mechanism" on page 65.
  - "Customize the WSDL Files" on page 68.

Customize the Notification Mechanism

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

Procedure: 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 Developer's Guide'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 Table 3 on page 66.

Start of procedure

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. Modified the element to include two custom tags: `debug` and `maxAttempts`. An example follows:

```
<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 of procedure

Next Steps

- Review the list of `modules.conf` options that you can change to customize the notification mechanism found at Table 3.
- Customize your GIS application further by:
  - “Windows Service Setting” on page 59.
  - “LCA Configuration” on page 60.
  - “SDK Server Connections for GIS SOAP” on page 62.
  - “Configure the Configuration Server SOAP Port” on page 63.
  - “Customize the Notification Mechanism” on page 65.
  - “Customize the WSDL Files” on page 68.

Table 3: 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.

Table 3: Statistical Notification Configuration Options (Continued)

Option Name	Value	Description
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.
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.

Procedure:
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 of procedure

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 of procedure

Customize the WSDL Files

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

Procedure:
Displaying the WSDL Files for GIS:SOAP

Purpose: To generate the WSDL files for each SDK Service.

Start of procedure

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 of procedure

Or you can:

Start of procedure

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 of procedure

Next Steps

- Customize your GIS application further by:
  - “Windows Service Setting” on page 59.
  - “LCA Configuration” on page 60.
  - “SDK Server Connections for GIS SOAP” on page 62.
  - “Configure the Configuration Server SOAP Port” on page 63.
  - “Customize the Notification Mechanism” on page 65.
  - “Customize the WSDL Files” on page 68.

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.

Procedure: 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 of procedure

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 of procedure

Next Steps

- Customize your GIS application further by:
  - "Windows Service Setting" on page 59.
  - "LCA Configuration" on page 60.
  - "SDK Server Connections for GIS SOAP" on page 62.
  - "Configure the Configuration Server SOAP Port" on page 63.
  - "Customize the Notification Mechanism" on page 65.
  - "Customize the WSDL Files" on page 68.

Procedure: 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 of procedure

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 of procedure

Next Steps

- Customize your GIS application further by:
  - “Windows Service Setting” on page 59.
  - “LCA Configuration” on page 60.
  - “SDK Server Connections for GIS SOAP” on page 62.
  - “Configure the Configuration Server SOAP Port” on page 63.
  - “Customize the Notification Mechanism” on page 65.
  - “Customize the WSDL Files” on page 68.

Procedure: 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.

Note: 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 of procedure

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

Note: 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 of procedure

Next Steps

- Customize your GIS application further by:
  - “Windows Service Setting” on page 59.
  - “LCA Configuration” on page 60.
  - “SDK Server Connections for GIS SOAP” on page 62.
  - “Configure the Configuration Server SOAP Port” on page 63.
  - “Customize the Notification Mechanism” on page 65.
  - “Customize the WSDL Files” on page 68.

Chapter

4 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. The chapter contains the following sections:

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.

Note: 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” on page 84).

- 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)” on page 74, and “Deployment for High Availability with Load Balancing” on page 80).

Note: 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 (either GIS:SOAP or GIS:GSAP).

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.

Note: 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” on page 80).

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:

- Chapter 2, “Installing and Uninstalling GIS,” on page 39.

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

- “Configuring the GIS Application Objects” on page 75
- “Configuring the HA Cluster for Manual Peer Discovery Support” on page 76
- “Configuring the Client Application” on page 79

Figure 4 on page 75 describes a GIS HA cluster with connections to clients.

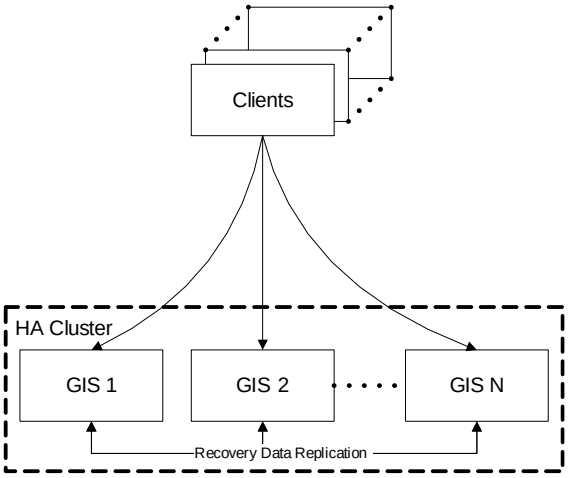

Figure 4: High Availability Cluster

Procedure: 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 Chapter 2, “Installing and Uninstalling GIS,” on page 39.

Start of procedure

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 of procedure

Next Steps

- Configure the HA cluster by following the steps in "Configuring the HA Cluster for Manual Peer Discovery Support" on page 76.

  Or:

- Configure the HA cluster by following the steps in "Configuring the HA Cluster for Automatic Peer Discovery Support" on page 77.

Note: 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 Chapter 2, "Installing and Uninstalling GIS," on page 39.
- Your GIS nodes must be configured for HA deployment using "Configuring the GIS Application Objects" on page 75.

Start of procedure

1. Confirm that an `ehcache.xml` configuration file is present in each GIS install directory (`webapps\gis\conf\` for GIS:SOAP, and `\config\` for GIS:GSAP).
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"/>
```

> Note: 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`

Table 4 shows the configuration of three GIS instances.

Table 4: 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 of procedure

Next Steps

- Configure your client application by following the steps in “Configuring the Client Application” on page 79.

Procedure: 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 Chapter 2, "Installing and Uninstalling GIS," on page 39.
- Your GIS nodes must be configured for HA deployment using "Configuring the GIS Application Objects" on page 75.
- You must have a multicast IP address in place.

Start of procedure

1. Confirm that an `ehcache.xml` configuration file is present in each GIS install directory (`webapps\gis\conf\` for GIS:SOAP, and `\config\` for GIS:GSAP). 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` (with important features bolded) 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="port=40001, 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>
```

Table 5 describes the important attributes from the `ehcache.xml` file.

Table 5: 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 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 of procedure

Next Steps

- Configure your client application by following the steps in “Configuring the Client Application” on page 79.

Procedure: 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 Chapter 2, “Installing and Uninstalling GIS,” on page 39.
- Your GIS nodes must be configured for HA deployment using “Configuring the GIS Application Objects” on page 75.
- Your `ehcache.xml` configuration file must be modified for your HA cluster using “Configuring the HA Cluster for Manual Peer Discovery Support” on page 76.

Start of procedure

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`.

`Note:` 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 of procedure

Next Steps

- To start and test GIS, Chapter 5, "Starting and Testing GIS," on page 91.

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.

Figure 5 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.

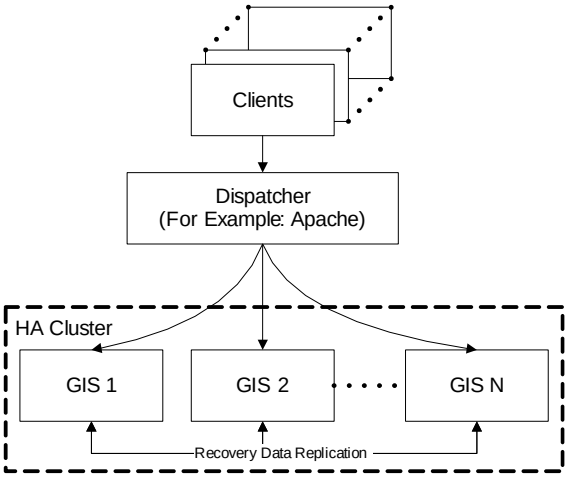

Figure 5: Deployment for High Availability with Load Balancing

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:

- Chapter 2, “Installing and Uninstalling GIS,” on page 39.

Also, the procedure described in the “Configuring the HA Cluster for Manual Peer Discovery Support” on page 76 must be completed.

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

- “Configuring Apache Dispatcher” on page 81
- “Configuring Tomcat for Apache Dispatcher Deployment” on page 83
- Step 2 of “Configuring the Client Application” on page 79

Note: The following procedures describe how to configure a load balanced HTTP cluster for SOAP. It is possible to configure a load balanced HTTP cluster for GSAP. This difference is, the Dispatcher is a front end GSAP instance that can be configured to manage either load balancing, or load balancing and client request handling.

Procedure: 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 Chapter 2, "Installing and Uninstalling GIS," on page 39.
- You must download the `mod_jk.so` Apache module matching your installed Apache version and place it in `Apache/modules/` directory.

Start of procedure

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 of procedure

Next Steps

- You must configure your Tomcat Application Server by following the steps in "Configuring Tomcat for Apache Dispatcher Deployment" on page 83

Procedure: 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 Chapter 2, "Installing and Uninstalling GIS," on page 39.
- You must configure the Apache Dispatcher using the procedure "Configuring Apache Dispatcher" on page 81.

Start of procedure

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>GISAXISServlet</servlet-name>
   <display-name>Apache-Axis Servlet</display-name>
   <servlet-class>com.genesyslab.gis.framework.GISAXISServlet
   </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 of procedure

Next Steps

- Configure your client application according to Step 2 of "Configuring the Client Application" on page 79.

Note: 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 either SOAP or GSAP.

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.

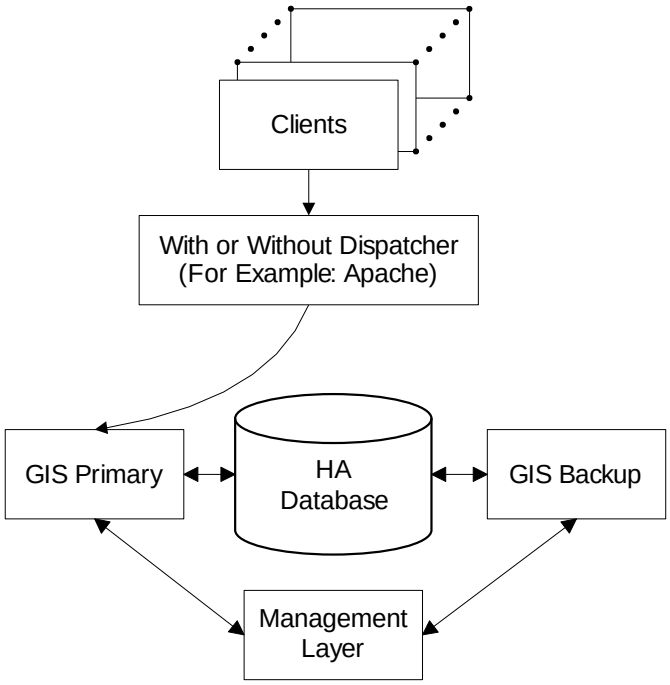

Figure 6: Alternative High Availability Deployment

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

- "Configuring and Installing GIS" on page 85.
- Chapter 2, "Installing and Uninstalling GIS," on page 39.

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

- "Configuring the GIS Application Objects" on page 75
- "Configuring the HA Cluster for Manual Peer Discovery Support" on page 76
- "Configuring the Client Application" on page 79

Procedure: 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" on page 24.

- 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.

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

Start of procedure

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.

   Note: 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 Step 4 above.) Set the `Redundancy Type` field to `Hot Standby`.

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

   ---

6. Repeat Step 5 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 Chapter 2, “Installing and Uninstalling GIS,” on page 39:

   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 of procedure

Next Steps

- Configure the server side according to “Configuring the Server Side” on page 87.

Procedure: Configuring the Server Side

Purpose: To configure the dispatcher to support high availability servers.

Prerequisites

- Complete the procedure “Configuring and Installing GIS” on page 85.
- You must be planning to use a dispatcher (See Figure 5 on page 81).

Start of procedure

Note: 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 of procedure

Next Steps

- Configure the client side using the procedure that suits your deployment type:
  - "Configuring the Client Side with Dispatcher" on page 88.
  - "Configuring the Client Side without Dispatcher" on page 88

Procedure: Configuring the Client Side with Dispatcher

Purpose: To configure the client side with a dispatcher.

Prerequisites

- Complete the procedure "Configuring and Installing GIS" on page 85.
- Complete the procedure "Configuring the Server Side" on page 87.

Start of procedure

1. Configure your dispatcher, as described in "Configuring Apache Dispatcher" on page 81.

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

End of procedure

Next Steps

- Start GIS, Chapter 5, "Starting and Testing GIS," on page 91

Procedure: Configuring the Client Side without Dispatcher

Purpose: To configure the client side without dispatcher.

Prerequisites

- Complete the procedure “Configuring and Installing GIS” on page 85.

Start of procedure

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

End of procedure

Next Steps

- Start GIS, Chapter 5, “Starting and Testing GIS,” on page 91.

GSAP High Availability Limitations

GSAP High Availability (HA) mechanism restores push mode event notification when the client reconnects to the server only after certain scenarios. The following list indicates in which instances restoration is supported:

- For versions prior to 7.5.002.01:
  - For Primary/Backup HA:
    - In the event of a network outage restoration will not occur.
    - In the event of a manual switchover restoration will not occur.
    - In the event of a primary server crash restoration will occur.
- For version 7.5.002.01 and later, but prior to 7.6.000.08 (first available 7.6 release):
  - For Primary/Backup HA:
    - In the event of a network outage restoration will not occur.
    - In the event of a manual switchover restoration will occur.
    - In the event of a primary server crash restoration will occur.
- For version 7.6.000.08 and later:
  - For Primary/Backup HA:
    - In the event of a network outage restoration will not occur.
    - In the event of a manual switchover restoration will occur.
    - In the event of a primary server crash restoration will occur.
  - For Clustered HA:
    - In the event of a network outage restoration will occur.
    - In the event of a primary server crash restoration will occur.

Note the following definitions:

Network Outage/Reconnect—The client loses the connection to the server and then reconnects, the server is running all the time. No switchover occurs.

Manual Switchover—A switchover is done using SCS, or the primary server is manually shut down.

Primary Crash—The primary server (for clustered mode, the server currently handling the particular client) terminates unexpectedly.

Chapter

5 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

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

Procedure: 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" on page 57.

Start of procedure

1. Locate the start scripts:
   - For GIS:SOAP installations, the start scripts are located in the `bin/` subdirectory under the top-level installation directory.
   - For GIS:GSAP installations, the start scripts are located in your GIS 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, AIX, and Tru64 UNIX platforms.
  - `run_gis_soap.bat` is a batch-file script used for startup on Windows platforms.
- For GIS:GSAP installations:
  - `startServer.sh` is the script used for startup on Solaris, HP-UX, Linux, AIX, and Tru64 UNIX platforms.
  - `startServer.cmd` is the 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" on page 92 procedure.
4. Run the script.

End of procedure

Next Steps

- You may need to modify the start script by following "Editing the Start Script" on page 92.

Procedure: 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" on page 91.

Start of procedure

1. Use a text editor to open and inspect the Windows, AIX, HP-UX, Linux, Solaris, or Tru64 UNIX 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 of procedure

Next Steps

- Start GIS, "Starting and Testing GIS" on page 93.

Procedure: 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 of procedure

1. Run the correct start script to start up GIS. See "Choosing Your Start Script" on page 91.
2. Access GIS using one of the following:
   - For GIS:SOAP installations, a web browser.
   - For GIS:GSAP installations, your compatible client application.
3. Follow one of the start procedures:
   - "Starting on UNIX-based Platforms (GIS:SOAP only)" on page 96.
   - "Starting GIS on Windows" on page 98.
   - "Starting GIS on a Web Application Server" on page 98.

End of procedure

Next Steps

- For GIS:SOAP, you may want to specify a licensing at startup, "Specifying Licensing at Startup (GIS:SOAP only)" on page 95.

Procedure: Configuring the License Section of the Option Tab (for GIS:SOAP or GIS:GSAP)

Purpose: To configure the "license Section" of the `Options` tab for GIS:SOAP and GIS:GSAP during configuration of a GIS `Application` object.

Note: You can also configure the license file parameter at startup, see Specifying Licensing at Startup (GIS:SOAP only), page 95.

Prerequisites

- You must have configured your application for GIS:SOAP or GIS:GSAP.

Notes: 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 only), page 95. If you do so, the command-line information takes priority over the "license-file" option settings. See Chapter 5, "Starting and Testing GIS," on page 91, for details.

Note: 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 of procedure

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" on page 117 as a guide to configuring additional settings in the `Options` tab's `license` section.

End of procedure

Next Steps

- Configure other sections of the Options Tab, see Appendix A, "Configuration Options," on page 103.

Procedure: Specifying Licensing at Startup (GIS:SOAP only)

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

Prerequisites

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

Note: You can also configure the license file location during configuration of the GIS `Application` object (see "Genesys Integration Server" on page 113).

Start of procedure

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.

   Notes: 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 of procedure

Next Steps

- "Verify the Status of GIS (GIS:SOAP Only)" on page 96.

Procedure:
Starting on UNIX-based Platforms (GIS:SOAP only)

Purpose: To start GIS:SOAP on UNIX-based platforms.

Prerequisites

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

Start of procedure

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).
   - `startServer.sh` (for GIS:GSAP installations).

> Note: 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 of procedure

Next Steps

- “Verify the Status of GIS (GIS:SOAP Only)” on page 96.
- “Stopping GIS on UNIX-based Platforms” on page 99.

Procedure:
Verify the Status of GIS (GIS:SOAP Only)

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

Prerequisites

- “Starting on UNIX-based Platforms (GIS:SOAP only)” on page 96.

Start of procedure

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`

   Note: 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 of procedure

Next Steps

- “Stopping GIS on UNIX-based Platforms” on page 99.

Procedure:
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 of procedure

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`.

Using the batch file (for GIS:GSAP):

1. Navigate to your GIS installation directory.
2. Double-click `startServer.cmd`.

End of procedure

Next Steps

- “Stopping GIS on Windows” on page 100.

Procedure:
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 of procedure

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 of procedure

Next Steps

- “Stopping on a Web Application Server” on page 100.

Stopping GIS

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

Procedure: 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 of procedure

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`
   - (For GIS:GSAP:) `cd <GIS_installation_directory>`
3. Issue the `./shutdown.sh` command (for GIS:SOAP) or the `stopServer.sh` command (for GIS:GSAP).

End of procedure

Procedure:
Stopping GIS on Windows

Purpose: To stop GIS on Windows 2000 or Windows 2003 by using either a Start menu shortcut or a batch file.

Prerequisites

- You must be running GIS on Windows 2000 or Windows 2003.

Start of procedure

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`.

Using the batch file (for GIS:GSAP):

1. Navigate to your GIS installation directory.
2. Double-click `stopServer.cmd`.

End of procedure

Procedure:
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 of procedure

1. Consult your application server's documentation.

End of procedure

Procedure:
Using Management Layer to Start and Stop GIS

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

Prerequisites

- GIS must be installed.

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

Start of procedure

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 of procedure

Notes: For more information, see *Framework 7.x Solution Control Interface Help*. To view the Help, open SCI, and click `Help`. For more information on Management Layer topics, see the *Framework 7.x Management Layer User's Guide*.

Appendix

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. The chapter contains the following sections:

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

Agent Interaction Service

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" on page 104.
- (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`

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`

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`

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

reroute

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

Specifies whether the reroute capability is enabled (`true`) or disabled (`false`).

reroute-location

Default Value: `<location>`
Valid Values: *switch1,...,switchN*

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>*

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

attempts-max

Default Value:`10`
Valid Value: *<Any positive integer>*

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>`

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

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

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`

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`

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)

Specifies how to write entries in log files.

multimedia Section

chat-addp-protocol

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

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>*

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

`Note:` Not connected to routing strategy.

---

collaboration-workbin

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

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`

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>*

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>*

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

`Note:` Not connected to routing strategy.

---

email-quote-char

Default Value: `>`
Valid Values: *<Any string value>*

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>*

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.

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

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`

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.

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>*

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>*

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`

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>*

Use to define each line of the signature.

prefix

Default Value: `"--"`
Valid Values: *<Any string>*, *<empty string>*

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

voice Section

a4400-custom-substitute-mode

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

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

database

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

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

`all`—Any voice call uses the database.

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

`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>*

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`

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`

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`

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`

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`

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.

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 Interaction Service

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`

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).

Used to connect to the Interaction Server to be able to perform ad-hoc features.

log Section

omsdk-console

Default Value: `info`
Valid Values: `false, debug, info, warn, error, fatal`

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]

Specifies how to store traces to a file.

<*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).

omsdk-msgsrv

Default Value: `off`
Valid Values: `off, debug, info, warn, fatal`

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`

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 coverts 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`

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

use-lca

Default Value: `false`
Valid Values: `false`

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`

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`.

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

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 or GIS:GSAP), page 94.

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” on page 113
- “core-services Section” on page 114
- “license Section” on page 117
- “log Section” on page 118
- “session-services Section” on page 123
- “stat-services Section” on page 124

To configure GSAP options use the following information as a guide:

- “[GSAP.General] Section” on page 124
- “[GSAP.KeepAlive] Section” on page 126
- “[GSAP.LoadBalancing] Section” on page 127

For additional details about configuring GSAP.LoadBalancing options, see “GSAP Load Balancing Configuration” on page 61.

ail-services Section

agent-wait-status-stable

Default Value: `100`
Valid Values: Integer values from `0` to `1000`.
Changes Take Effect: After restart.

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 from `0`–`60`.
Changes Take Effect: After restart.

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.

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.

timeout

Default Value: **`30`**
Valid Values: Any positive integer.
Changes Take Effect: After restart.

Specifies the timeout value, in seconds, of requests sent to back-end servers. For example, loading `cfg` data.

receive-self-user-event

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

If set to `true`, GIS will send to its clients the same UserEvent message that it sent to the T-Server.

core-services Section

events-adapter-soap-commons-sender-enabled

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

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.

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.

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.

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.

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.

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.

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.

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: `300`
Valid Values: Integer values from `0` to `500`.
Changes Take Effect: Immediately.

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.

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.

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.

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.

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.

events-thread-priority

Default Value: 7
Valid Values: Integer values from 1 to 10.
Changes Take Effect: After restart.

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.

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.

Notes:

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.

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_statservice-block-size

Default Value: `1`

Valid Values: Any positive value.

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_configurationservice-block-size

Default Value: `1`

Valid Values: Any positive value.

Specifies the 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*

Addresses of FlexLM license servers.

log Section

Note: 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.

For GIS:GSAP deployments, the logs for Agent Interaction Services and Open Media Interaction Services are controlled by the options set in the related Services `Application` object.

all

Default Value: `network, ../logs/all.log, stdout`
Valid Values: `network, ../logs/all.log, all.log, stdout`

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

Note: This option affects only SOAP deployments of GIS.

Buffering

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

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.

Note: This option affects only SOAP deployments of GIS.

debug

Default Value: [none]
Valid Values: `network, ../logs/all.log, all.log, stdout`

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

Note: This option affects only SOAP deployments of GIS.

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.

Determines whether log files will be deleted, and if so, the deletion interval (set as a maximum number of log files or retention days).

`Note:` This option affects only SOAP deployments of GIS.

---

lb-console

Default Value: `info`
Valid Values: `off, debug, info, warn, fatal`

Specifies level and size of load-balancing traces that appear on the internal console. If this option is not specified, then load-balancing traces are not logged to the MessageServer.

`Note:` This option affects only GSAP deployments of GIS.

---

lb-file

Default Value: `info, gis-gsap-lb.log, 20, 3`
Valid Values: <*level*>,<*file name*>, <*max file size*>, <*max file number*>

Used to write traces of the load-balancing engine to the file. If this option is not specified, there is no load-balancing logging to the file.

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

`Note:` This option affects only GSAP deployments of GIS.

---

lb-msgsrv

Default Value: `off`
Valid Values: `off, debug, info, warn, error, fatal`

Specifies level and size of the load-balancing engine for the MessageServer centralized log. If this option is not specified, load-balancing logging to MessageServer does not occur.

Note: This option affects only GSAP deployments of GIS.

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.

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.

Note: This option affects only SOAP deployments of GIS.

prop-console

Default Value: info
Valid Values: off, debug, info, warn, fatal

Specifies level and size of this server's protocol traces for the internal console. If this option is not specified, load-balancing logging to console does not occur.

Note: This option affects only GSAP deployments of GIS.

prop-file

Default Value: info, gis-gsap.log, 20, 3
Valid Values: <*level*>,<*file name*>, <*max file size*>, <*max file number*>

Used to write traces from this server to the file. If this option is not specified, this server does not log to the file.

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

Note: This option affects only GSAP deployments of GIS.

prop-msgsrv

Default Value: `off`
Valid Values: `off, debug, info, warn, fatal`

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.

Note: This option affects only GSAP deployments of GIS.

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.

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.

Note: This option affects only SOAP deployments of GIS.

standard

Default Value: [none]
Valid Values: `network, ../logs/all.log, all.log, stdout`

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

Note: This option affects only SOAP deployments of GIS.

trace

Default Value: [none]
Valid Values: `network, ../logs/all.log, all.log, stdout`

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

Note: This option affects only SOAP deployments of GIS.

verbose

Default Value: `trace`
Valid Values: `all, debug, standard, trace`

Sets the target log's level.

Note: This option affects only SOAP deployments of GIS.

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

This parameter specifies how long the client must wait between `retrieveSubscribedStatistic` requests.

error_check

Default Value: `true`

Valid Values: `true` or `false`

This option establishes whether GIS uses error checking during statistics subscriptions. To disable error checking, set this value to `false`.

Note: 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:

Use this option to set the maximum number of statistic events that a client application can retrieve for one statistic.

[GSAP.General] Section

ha-session-ttl

Default Value: `86400000` (24 hours)

Valid Value: Any positive integer.

Note: This option applies to the HA-DB architecture only.

Specifies the time interval (milliseconds) that a session record will be saved in the database. Specifically, if the difference between the current time (ct) and the last-updated time (lut) is greater than ha-session-ttl ([ct-lut] > [ha-session-ttl]), the record is inactive and it will be deleted.

Note: Inactive-session records are deleted to manage the database size.

max.param.count

Default Value: `200`
Valid Value: Any positive integer.

Maximum size of array for deserialized objects when deserialization takes place.

nat-mode

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

When set to `false`, GIS.GSAP can not work with clients through NAT.

When set to `true`, GIS.GSAP is able to work with clients through NAT with following drawbacks:

- Load Balancing will be disabled for this server, even if `lb-cluster-servers` is set.
- Clients using .NET proxy servers prior to version 7.6.000.01, will encounter an error during the handshake phase and disconnect from the server.

Note: There are no compatibility issues with any versions of the Java proxy servers.

Note: The `nat-mode` option is not included in the GIS configuration template, and must be added manually to your GIS configuration options.

server.backlog

Default Value: `1000`
Valid Values: Any positive integer.

Specifies the maximum number of client connections in the server-socket backlog-queue. Any client's connection request is refused if the queue is full.

server.clientsocket.enablekeepalive

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

Specifies whether to enable or disable the `SO_KEEPALIVE` parameter for client sockets.

server.clientsocket.enabletcpnodelay

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

Specifies whether to enable or disable the TCP_NODELAY parameter for client sockets.

server.clientsocket.sendbufsize

Default Value: 524288
Valid Values: Any positive integer.

The SO_SNDBUF value is used by the platform networking code as a hint for the size to set the underlying network Input/Output buffers.

server.clientsocket.sotimeout

Default Value: 10000
Valid Values: Any positive integer.
Changes Take Effect: After Desktop Server is restarted.

Specifies whether to enable or disable SO_TIMEOUT with the specified timeout, in milliseconds, for the client socket. With this option set to a non-zero timeout, a read() call on the InputStream associated with this socket will block for only this amount of time.

The timeout must be greater than zero; a timeout of zero is interpreted as an infinite timeout.

server.handlers.max

Default Value: 2000
Valid Values: Any positive integer.

Specifies the peak expected server load (that is, the maximum number of expected simultaneous requests).

If the client applications are using the pull-event mode, this value should be equal to, or greater than, the number of client applications connected to GIS.

Note: If the number of simultaneous requests exceeds this value, client requests will be dropped.

server.handlers.recommended

Default Value: 200
Valid Values: Any positive integer.

Specifies the average expected server load (that is, the average number of expected simultaneous requests).

[GSAP.KeepAlive] Section

handler-daemon-audit-interval

Default Value: 5000
Valid Values: Any positive integer.

Threshold value, in milliseconds, for the audit interval during which the handler tracker daemon thread tracks and releases resources used by disconnected clients.

handler-daemon-inactivity-interval

Default Value: `1000`
Valid Values: Any positive integer.

Sleep interval, in milliseconds, for the handler tracker daemon thread. This thread is inactive during the specified interval.

ping-daemon-audit-interval

Default Value: `3000`
Valid Values: Any positive integer.

Waiting time threshold, in milliseconds, for Ping response messages from the client. If no response, the client is either disconnected or Pinged once more (if `ping-times-threshold` is not exceeded).

ping-daemon-inactivity-interval

Default Value: `500`
Valid Values: Any valid positive integer.

Sleep interval, in milliseconds, for the Ping daemon thread. This thread is inactive during the specified interval.

ping-times-threshold

Default Value: `3`
Valid Values: Any valid positive integer.

Threshold, in milliseconds, for the amount of time the client connection will be Pinged before disconnection from the server.

session-timeout-threshold

Default Value: `3600000`
Valid Values: Any positive integer.

Threshold value, in milliseconds, for session inactivity between client and server. After this time slot, a Ping request should be sent from server to client.

[GSAP.LoadBalancing] Section

lb-clientsocket-enabletcpnodelay

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

To enable or disable `TCP_NODELAY` for the `LoadBalancer` component's client socket.

lb-clientsocket-sotimeout

Default Value: `5000`
Valid Value: Any valid positive integer.

To enable or disable the socket timeout (`SO_TIMEOUT`) with the specified timeout, in milliseconds, for client socket of a `LoadBalancer` component.

lb-cluster-servers

Default Value: [none]
Valid Value: A sequence of <*host*>:<*port*> elements, separated by a comma.
For example:92.168.83.77:1200, 127.0.0.1:1200

Each <*host*>:<*port*> element stands for the address of one server in the cluster.

If there is no value, the server will not try to connect to other GIS:GSAP connectors.

lb-connector-exchange-interval

Default Value: `5000`
Valid Value: Any positive integer

Interval, in milliseconds, before load information updates are sent to remote GIS:GSAP connectors in a cluster.

lb-connector-process-interval

Default Value: `30`
Valid Value: Any positive integer number.

Sleep interval, in milliseconds, for `LBConnector` component socket listening thread. This thread is in idle stage for a determined time (Genesys recommends not less than `20ms`).

lb-load-threshold

Default Value: `95.50`
Valid Value: Any positive integer (float).

Specifies the threshold load value (as a percentage) for the current GIS:GSAP connector. This value specifies when to start redirecting connections to other servers in the cluster that have less load. A server will accept all client connections if the load on that server is less than the value specified by this option, even if the other servers in the cluster have a zero load value.

`Note`: It can be useful to give the primary server a lower value for this option, so that resources are free to handle new connections.

---

lb-load-delta-threshold

Default Value: `20`
Valid Value: Any positive integer less than or equal to 100.

Specifies the delta threshold value (as a percentage), after which updates are sent to remote GIS:GSAP connectors in cluster without waiting for the `lb-connector-process-interval`.

lb-max-connections-per-server

Default Value: `100`
Valid Value: Any positive integer, up to 1000.

Maximum number of connections for each server in the cluster. This value is used by the loading algorithm. This value does not represent the limit of accepted connection; it is one of the values that are used by the load calculation algorithm. The real limit of connection is set by the `DotNetServer.General:app.worker.threads` option.

> Note: It can be useful to give the primary server a lower value for this option, so that resources are free to handle new connections.

lb-max-transfer-count

Default Value: `2`
Valid Value: Any positive integer.

Maximum transfer count threshold for client. After this threshold, the client connection is accepted regardless of the current server load.

lb-port

Default Value: `1201`
Valid Value: Any valid port number.

Specifies a local port for the `LoadBalancer` component. If no port is specified, LoadBalancer does not start.

lb-process-interval

Default Value: `30`
Valid Value: Any positive integer.

Sleep interval, in milliseconds, for `LoadBalancer` component socket listening thread. This thread is in idle stage for a determined time (Genesys recommends not less than `20ms`).

Index

E

F

G

H

I

J

K

L

M

N

O

P

S

T

U

V

W

X