



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

GIS Deployment Guide

Customizing the GIS Environment

12/14/2025

Contents

- 1 Customizing the GIS Environment
 - 1.1 Windows Service Setting
 - 1.2 LCA Configuration
 - 1.3 Tuning the Heap Size Values
 - 1.4 SDK Server Connections for GIS SOAP
 - 1.5 Configure the Configuration Server SOAP Port
 - 1.6 Customize the Notification Mechanism
 - 1.7 The HTTP Connection Element
 - 1.8 Customize the WSDL Files
 - 1.9 GZIP Compression

Customizing the GIS Environment

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

Windows Service Setting

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

Warning

Do not change this setting; changes are not supported.

LCA Configuration

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

Important

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

Tuning the Heap Size Values

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

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

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

Tuning Heap Size Values

Purpose

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

Start

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

End

SDK Server Connections for GIS SOAP

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

Configuration SDK Server Connections

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

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

Configure the Configuration Server SOAP Port

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

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

Important

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

Configuring a SOAP Port for a Master Configuration Server

Purpose

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

Start

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

End Next Steps

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

Configuring a SOAP Port for a CS Proxy

Purpose

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

Start

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

Important

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

End Next Steps

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

Customize the Notification Mechanism

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

Customizing the Notification Mechanism for GIS SOAP

Purpose

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

Prerequisites

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

Start

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

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

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

End Next Steps

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

Statistical Notification Configuration Options

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

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

The HTTP Connection Element

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

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

Setting the HTTP Connector Thread Number for GIS SOAP

Purpose

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

Start

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

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

End

Customize the WSDL Files

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

Displaying the WSDL Files for GIS SOAP

Purpose

To generate the WSDL files for each SDK Service.

Start

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

End

Or you can:

Start

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

End

Next Steps

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

- [Customize the WSDL Files.](#)

GZIP Compression

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

Enabling GZIP Compression for .NET

Purpose

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

Prerequisites

- Requires .NET 2.0 or higher.

Start

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

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

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

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

End

Next Steps

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

Enabling GZIP Compression for Java

Purpose

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

Start

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

```
AcceptGZIP = true
```

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

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

End

Next Steps

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

Enabling GZIP Compression using Web Server Features

Purpose

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

Important

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

Start

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

Important

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

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

End Next Steps

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