



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

High-Availability Deployments of GIS

12/17/2025

Contents

- 1 High-Availability Deployments of GIS
 - 1.1 Overview
 - 1.2 Deployment for High Availability (No Load Balancing)
 - 1.3 Deployment for High Availability with Load Balancing
 - 1.4 Alternative Deployment for High Availability

High-Availability Deployments of GIS

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

Overview

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

Important

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

logout-on-disconnect is set to true

.

GIS HA may be setup using either:

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

Important

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

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

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

Important

Machines hosting GIS nodes must have their clocks synchronized.

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

Deployment for High Availability (No Load Balancing)

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

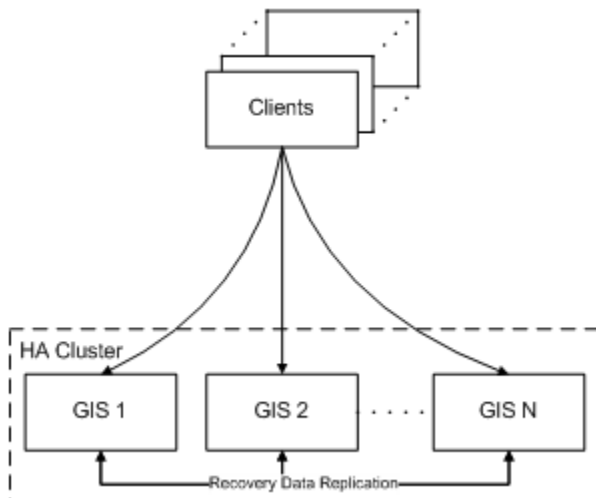
A standard HA deployment uses a cluster of GIS nodes, where each node is a GIS instance that has been fully configured and installed as described in:

- [Installing and Uninstalling GIS](#).

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

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

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



High Availability Cluster

Configuring the GIS Application Objects

Purpose

To configure your GIS application objects for High Availability deployment.

Prerequisites

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

Start

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

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

Other considerations you should keep in mind include:

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

End

Next Steps

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

Or:

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

Important

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

Automatic peer discovery is:

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

Configuring the HA Cluster for Manual Peer Discovery Support

Purpose

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

Prerequisites

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

Start

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

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

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

Important

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

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

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

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

Configuration of Three GIS Instances (Example)

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

End Next Steps

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

Configuring the HA Cluster for Automatic Peer Discovery Support

Purpose

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

Prerequisites

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

Start

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

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

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

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

<cacheManagerPeerListenerFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"
properties="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>
```

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

Important ehcache.xml Attributes

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

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

End

Next Steps

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

Configuring the Client Application

Purpose

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

Prerequisites

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

Start

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

Important

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

End

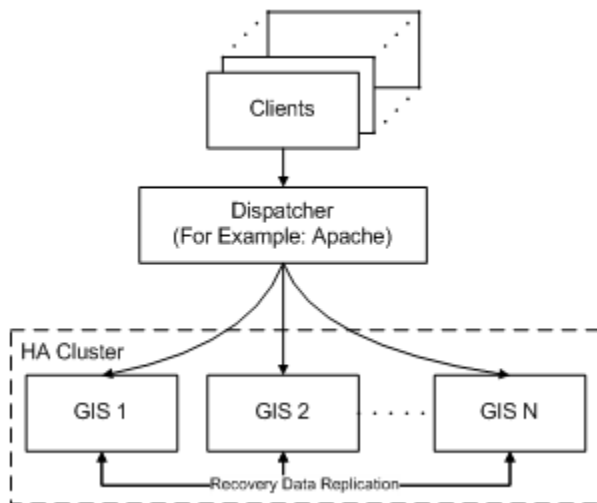
Next Steps

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

Deployment for High Availability with Load Balancing

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

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



Deployment for High Availability with Load Balancing

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

- [Installing and Uninstalling GIS.](#)

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

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

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

Important

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

Configuring Apache Dispatcher

Purpose

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

Prerequisites

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

Start

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

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

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

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

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

Warning

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

End Next Steps

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

Configuring Tomcat for Apache Dispatcher Deployment

Purpose

To configure Tomcat for Apache Dispatcher.

Prerequisites

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

Start

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

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

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

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

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

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

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

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

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

End Next Steps

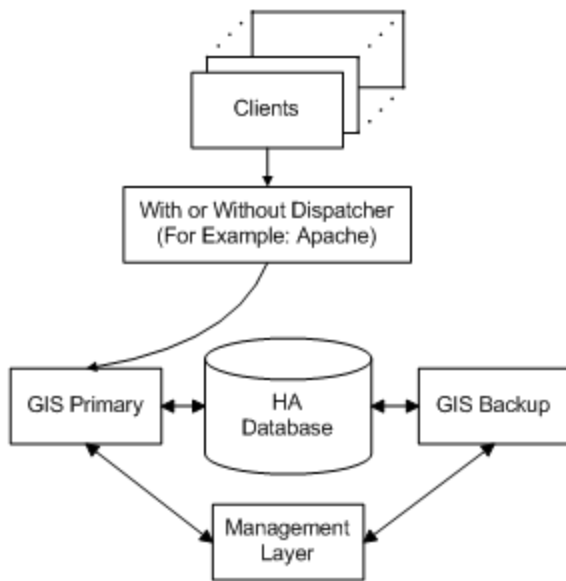
- Configure your client application according to See For GIS:SOAP, using an HTTP Dispatcher, you must enable cookies in your client program. Ensure that the UseCookieContainer option is set to true. of [Configuring the Client Application](#).

Important

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

Alternative Deployment for High Availability

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



Alternative High Availability Deployment

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

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

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

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

Configuring and Installing GIS

Purpose

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

Prerequisites

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

Important

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

Start

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

Important

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

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

Important

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

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

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

End Next Steps

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

Configuring the Server Side

Purpose

To configure the dispatcher to support high availability servers.

Prerequisites

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

Start

Important

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

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

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

Warning

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

End Next Steps

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

Configuring the Client Side with Dispatcher

Purpose

To configure the client side with a dispatcher.

Prerequisites

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

Start

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

Important

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

End Next Steps

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

Configuring the Client Side without Dispatcher

Purpose

To configure the client side without dispatcher.

Prerequisites

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

Start

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

End Next Steps

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