`embedded-glassfish:stop` Goal

This goal stops the server. You can set the parameters described in the following table.

TABLE 7 `embedded-glassfish:stop` Parameters

Parameter	Default	Description
serverID	`maven`	(optional) The ID of the server to stop.

Using the EJB 3.1 Embeddable API with Embedded GlassFish Server

The EJB 3.1 Embeddable API is designed for unit testing of EJB modules. You must use this API with a pre-installed, nonembedded GlassFish Server instance. However, you can take advantage of the embedded GlassFish Server's ease of use by referencing the nonembedded GlassFish Server instance with the `glassfish-embedded-static-shell.jar` file.

Embedded GlassFish Server is not related to the EJB 3.1 Embeddable API, but you can use these APIs together.

The Maven plug-in does not apply to embeddable EJB applications. However, you can use Maven with the POM file shown in "Using Maven with the EJB 3.1 Embeddable API and Embedded GlassFish Server" on page 35.

The EJB 3.1 Embeddable API is described in Java Specification Request (JSR) 318 (`http://jcp.org/en/jsr/detail?id=318`). An `ejb-embedded` sample is included in the samples available at Java EE 6 Downloads (`http://java.sun.com/javaee/downloads/index.jsp`) or Code Samples (`http://java.sun.com/javaee/reference/code/index.jsp`).

▼ To Use the EJB 3.1 Embeddable API with Embedded GlassFish Server

1 **To specify GlassFish Server as the Container Provider, include `glassfish-embedded-static-shell.jar` in the class path of your embeddable EJB application.**

See "Setting the Class Path" on page 12 and Section 22.3.3 of the EJB 3.1 Specification, *Embeddable Container Bootstrapping*.

2 **Configure any required resources.**

For more information about configuring resources, see the Administration Console Online Help or Part III, "Resources and Services Administration," in *Oracle GlassFish Server 3.0.1 Administration Guide*. The `jdbc/__default` Java DB database is preconfigured with all distributions of GlassFish Server.

3 **Invoke one of the `createEJBContainer` methods.**

Note – *Do not* deploy your embeddable EJB application or any of its dependent Java EE modules before invoking one of the `createEJBContainer` methods. These methods perform deployment in the background and do not load previously deployed applications or modules.

---

4 **To change the Instance Root Directory, set the `org.glassfish.ejb.embedded.glassfish.instance.root` system property value by using the `createEJBContainer(Map<?, ?> properties)` method.**

The default Instance Root Directory location is *as-install*`/domains/domain1`. This system property applies only to embeddable EJB applications used with nonembedded GlassFish Server.

5 **Close the EJB container properly to release all acquired resources and threads.**

Using Maven with the EJB 3.1 Embeddable API and Embedded GlassFish Server

When using Maven with the EJB 3.1 Embeddable API and Embedded GlassFish Server, you cannot use the features of the Maven plug-in. You must start and stop Embedded GlassFish Server manually or programmatically outside of Maven.

EXAMPLE 13 Maven POM File for Using the EJB 3.1 Embeddable API with Embedded GlassFish Server

This example shows a POM file for configuring Maven to use the EJB 3.1 Embeddable API with Embedded GlassFish Server.

```
<!--
Line breaks in the following element are for readability purposes only
-->
<project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.glassfish</groupId>
    <artifactId>maven-glassfish-plugin-tester</artifactId>
    <version>3.0-74b</version>
    <name>Maven test</name>
    <dependencies>
        <dependency>
```

EXAMPLE 13 Maven POM File for Using the EJB 3.1 Embeddable API with Embedded GlassFish Server *(Continued)*

```
            <groupId>org.glassfish.extras</groupId>
            <artifactId>glassfish-embedded-all</artifactId>
            <version>3.0</version>
        </dependency>
<!--
        The javaee-api is stripped of any code and is just used to compile your
        application. The scope provided in Maven means that it is used for compiling,
        but is also available when testing. For this reason, the javaee-api needs to
        be below the embedded Glassfish dependency. The javaee-api can actually be
        omitted when the embedded Glassfish dependency is included, but to keep your
        project Java-EE 6 rather than GlassFish 3, specification is important.
-->
        <dependency>
            <groupId>javax</groupId>
            <artifactId>javaee-api</artifactId>
            <version>6.0</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>
    <pluginRepositories>
        <pluginRepository>
            <id>maven2-repository.dev.java.net</id>
            <name>Java.net Repository for Maven</name>
            <url>http://download.java.net/maven/glassfish/</url>
        </pluginRepository>
    </pluginRepositories>
</project>
```

Changing Log Levels in Embedded GlassFish Server

To change log levels in Embedded GlassFish Server, you can follow the steps in this section or you can use the Embedded Server API as shown in Example 9. For more information about GlassFish Server logging, see Chapter 7, "Administering the Logging Service," in *Oracle GlassFish Server 3.0.1 Administration Guide*.

▼ To Change Log Levels in Embedded GlassFish Server

1 **Ensure that you have write permission access to the `$JAVA_HOME/jre/lib/logging.properties` file.**

2 **Use a text editor to edit the `$JAVA_HOME/jre/lib/logging.properties` file.**

3 **Change the `java.util.logging.ConsoleHandler.level` log level to `FINE` or `FINEST`.**

4 **Add GlassFish Server log levels to the end of the file and adjust them as necessary.**

For example, add `javax.enterprise.system.tools.deployment.level=FINE`.