

Deprecated, Unsupported, and Obsolete Options

Options in [Table 1-2](#) are deprecated or no longer supported, or are obsolete and are ignored.

TABLE 1-2 Deprecated, Unsupported, and Obsolete Options for `asadmin` and Subcommands

Option	Affected Subcommands
<code>--acceptlang</code>	Unsupported for the <code>create-virtual-server</code> subcommand.
<code>--acls</code>	Unsupported for the <code>create-virtual-server</code> subcommand.
<code>--adminpassword</code>	Unsupported for all relevant subcommands. Use <code>--passwordfile</code> instead.
<code>--autohadb</code>	Obsolete for the <code>create-cluster</code> subcommand.
<code>--blockingenabled</code>	Unsupported for the <code>create-http-listener</code> subcommand.
<code>--configfile</code>	Unsupported for the <code>create-virtual-server</code> subcommand.
<code>--defaultobj</code>	Unsupported for the <code>create-virtual-server</code> subcommand.
<code>--defaulttvs</code>	Deprecated for the <code>create-http-listener</code> subcommand. Use <code>--default-virtual-server</code> option instead.
<code>--devicesize</code>	Obsolete for the <code>create-cluster</code> subcommand.
<code>--haadminpassword</code>	Obsolete for the <code>create-cluster</code> subcommand.
<code>--haadminpasswordfile</code>	Obsolete for the <code>create-cluster</code> subcommand.
<code>--haagentport</code>	Obsolete for the <code>create-cluster</code> subcommand.
<code>--haproperty</code>	Obsolete for the <code>create-cluster</code> subcommand.
<code>--hosts</code>	Obsolete for the <code>create-cluster</code> subcommand.
<code>--mime</code>	Unsupported for the <code>create-virtual-server</code> subcommand.
<code>--password</code>	Unsupported for all remote subcommands. Use <code>--passwordfile</code> instead.
<code>--path</code>	Unsupported for the <code>create-domain</code> subcommand. Use <code>--domaindir</code> instead.
<code>--portbase</code>	Obsolete <i>only</i> for the <code>create-cluster</code> subcommand. This option is still valid in other subcommands such as <code>create-domain</code> , <code>create-instance</code> , and <code>create-local-instance</code> .
<code>--resourcetype</code>	Unsupported for all relevant subcommands. Use <code>--restype</code> instead.

Applications That Use Java DB

The directory location of Java DB in GlassFish Server 3.1 has changed from its location in previous installations. Suppose that you have deployed applications that use Java DB databases in your previous server installation, and you upgrade your existing installation to GlassFish Server 3.1. If you run the `asadmin start -database` command and successfully start Java DB, you could run into problems while trying to run applications that were deployed on your previous server installation.

To solve this problem, you can copy the `databases` directory from your previous installation to `as-install/databases`. Make sure the database is not running when you do this.

Alternatively, you can perform these steps:

1. After upgrade, start GlassFish Server 3.1.
2. Use the `asadmin start -database` command with the `--dbhome` option pointing to the `databases` directory in the older version of Java DB. For example:

```
asadmin start-database --dbhome c:\glassfish\databases
```

Applications That Use Persistence

GlassFish Server 3.1 and 3.0.1, and Sun GlassFish Enterprise Server v3 and v3 Prelude use the persistence provider EclipseLink, while earlier versions used TopLink Essentials.

An application that uses the container to create an `EntityManager` or `EntityManagerFactory` and that used Toplink Essentials as its provider will work in GlassFish Server 3.1. The container creates an `EntityManager` if the application uses the `@PersistenceContext` annotation to inject an `EntityManager`, as in the following example:

```
@PersistenceContext  
EntityManager em;
```

The container creates an `EntityManagerFactory` if the application uses the `@PersistenceUnit` annotation to inject an `EntityManagerFactory`, as in the following example:

```
@PersistenceUnit  
EntityManagerFactory emf;  
  
EntityManager em = emf.createEntityManager();
```

When the application is loaded, GlassFish Server 3.1 will translate the provider to EclipseLink and will also translate `toplink.*` properties in the `persistence.xml` to corresponding EclipseLink properties. (The actual `persistence.xml` file remains unchanged.)