Java Web Start

Java Web Start allows your application client to be easily launched and automatically downloaded and updated. It is enabled for all application clients by default. For more information, see "Using Java Web Start" on page 198.

Application Client JAR File

In GlassFish Server 3.1, the downloaded appclient JAR file is smaller than in previous releases, with dependent classes in separate JAR files. When copying the downloaded appclient to another location, make sure to include the JAR files containing the dependent classes as well. You can also use the `asadmin get-client-stubs` command to retrieve the appclient and all associated application JAR files and place them in another location.

Developing Clients Using the ACC

This section describes the procedure to develop, assemble, and deploy client applications using the ACC.

The following topics are addressed here:

- "To Access an EJB Component From an Application Client" on page 195
- "To Access a JMS Resource From an Application Client" on page 197
- "Using Java Web Start" on page 198
- "Using the Embeddable ACC" on page 208
- "Running an Application Client Using the `appclient` Script" on page 209
- "Using the `package-appclient` Script" on page 210
- "The `client.policy` File" on page 210
- "Using RMI/IIOP Over SSL" on page 210
- "Connecting to a Remote EJB Module Through a Firewall" on page 212
- "Specifying a Splash Screen" on page 212
- "Setting Login Retries" on page 213
- "Using Libraries with Application Clients" on page 213

▼ To Access an EJB Component From an Application Client

1. **In your client code, reference the EJB component by using an `@EJB` annotation or by looking up the JNDI name as defined in the `ejb-jar.xml` file.**

   For more information about naming and lookups, see "Accessing the Naming Context" on page 253.

If load balancing is enabled as in Step 7 and the EJB components being accessed are in a different cluster, the endpoint list must be included in the lookup, as follows:

`corbaname:`*host1*`:`*port1*`,`*host2*`:`*port2*`,`*...*`/NameService#ejb/`*jndi-name*

2 **Define the `@EJB` annotations or the `ejb-ref` elements in the `application-client.xml` file. Define the corresponding `ejb-ref` elements in the `glassfish-application-client.xml` file.**

For more information on the `glassfish-application-client.xml` file, see "The glassfish-application-client.xml file" in *GlassFish Server Open Source Edition 3.1 Application Deployment Guide*. For a general explanation of how to map JNDI names using reference elements, see "Mapping References" on page 260.

3 **Deploy the application client and EJB component together in an application.**

For more information on deployment, see the *GlassFish Server Open Source Edition 3.1 Application Deployment Guide*. To get the client JAR file, use the `--retrieve` option of the `asadmin deploy` command.

To retrieve the stubs and ties generated during deployment, use the `asadmin get-client-stubs` command. For details, see the *GlassFish Server Open Source Edition 3.1 Reference Manual*.

4 **Ensure that the client JAR file includes the following files:**

- A Java class to access the bean.
- `application-client.xml` - (optional) Java EE application client deployment descriptor.
- `glassfish-application-client.xml` - (optional) GlassFish Server specific client deployment descriptor. For information on the `glassfish-application-client.xml` file, see "The glassfish-application-client.xml file" in *GlassFish Server Open Source Edition 3.1 Application Deployment Guide*.
- The `MANIFEST.MF` file. This file contains a reference to the `main` class, which states the complete package prefix and class name of the Java client.

5 **Prepare the client machine.**

This step is not needed for Java Web Start. This step is not needed if the client and server machines are the same.

If you are using the `appclient` script, package the GlassFish Server system files required to launch application clients on remote systems using the `package-appclient` script, then retrieve the application client itself using the `asadmin get-client-stubs` command.

For more information, see "Using the `package-appclient` Script" on page 210 and the *GlassFish Server Open Source Edition 3.1 Reference Manual*.

6 **To access EJB components that are residing in a remote system, make the following changes to the `sun-acc.xml` file or the `appclient` script. This step is not needed for Java Web Start.**

- Define the target-server element's address and port attributes to reference the remote server machine and its ORB port. See "target-server" in *GlassFish Server Open Source Edition 3.1 Application Deployment Guide*.
- Use the -targetserver option of the appclient script to reference the remote server machine and its ORB port. For more information, see "Running an Application Client Using the appclient Script" on page 209.

To determine the ORB port on the remote server, use the asadmin get command. For example:

```
asadmin --host rmtsrv get server-config.iiop-service.iiop-listener.iiop-listener1.port
```

For more information about the asadmin get command, see the *GlassFish Server Open Source Edition 3.1 Reference Manual*.

7 To set up load balancing and failover of remote EJB references, define at least two target-server elements in the sun-acc.xml file or the appclient script. This step is not needed for Java Web Start.

If the GlassFish Server instance on which the application client is deployed participates in a cluster, the ACC finds all currently active IIOP endpoints in the cluster automatically. However, a client should have at least two endpoints specified for bootstrapping purposes, in case one of the endpoints has failed.

The target-server elements in the sun-acc.xml file specify one or more IIOP endpoints used for load balancing. The address attribute is an IPv4 address or host name, and the port attribute specifies the port number. See "client-container" in *GlassFish Server Open Source Edition 3.1 Application Deployment Guide*.

The --targetserver option of the appclient script specifies one or more IIOP endpoints used for load balancing. For more information, see "Running an Application Client Using the appclient Script" on page 209.

Next Steps

- For instructions on running the application client, see "Using Java Web Start" on page 198 or "Running an Application Client Using the appclient Script" on page 209.
- For more information about RMI-IIOP load balancing and failover, see Chapter 11, "RMI-IIOP Load Balancing and Failover," in *GlassFish Server Open Source Edition 3.1 High Availability Administration Guide*.

▼ To Access a JMS Resource From an Application Client

1 Create a JMS client.

For detailed instructions on developing a JMS client, see "Chapter 33: The Java Message Service API" in the *The Java EE 6 Tutorial* (http://download.oracle.com/javaee/6/tutorial/doc/).