

The value of each `redirect_n` property has two components which can be specified in any order:

- The first component, `from`, specifies the prefix of the requested URI to match.
- The second component, `url-prefix`, specifies the new URL prefix to return to the client. The `from` prefix is replaced by this URL prefix.

EXAMPLE 6-3 Redirecting a URL

This example redirects `from dummy` to `etude`:

```
<property name="redirect_1" value="from=/dummy url-prefix=http://etude"/>
```

Administering mod_jk

The Apache Tomcat Connector `mod_jk` can be used to connect the web container with web servers such as Apache HTTP Server. By using `mod_jk`, which comes with GlassFish Server, you can front GlassFish Server with Apache HTTP Server.

You can also use `mod_jk` directly at the JSP/servlet engine for load balancing. For more information about configuring `mod_jk` and Apache HTTP Server for load balancing with GlassFish Server 3.1 refer to [Chapter 8, “Configuring HTTP Load Balancing,” in *Oracle GlassFish Server 3.1 High Availability Administration Guide*](#).

The following topics are addressed here:

- [“To Enable mod_jk” on page 114](#)
- [“To Load Balance Using mod_jk and GlassFish Server” on page 116](#)
- [“To Enable SSL Between the mod_jk Load Balancer and the Browser” on page 118](#)
- [“To Enable SSL Between the mod_jk Load Balancer and GlassFish Server” on page 119](#)

▼ To Enable mod_jk

You can front GlassFish Server with Apache HTTP Server by enabling the `mod_jk` protocol for one of GlassFish Server's network listeners, as described in this procedure. A typical use for `mod_jk` would be to have Apache HTTP Server handle requests for static resources, while having requests for dynamic resources, such as servlets and JavaServer Pages (JSPs), forwarded to, and handled by the GlassFish Server back-end instance.

When you use the `jk-enabled` attribute of the network listener, you do not need to copy any additional JAR files into the `/lib` directory. You can also create JK connectors under different virtual servers by using the network listener attribute `jk-enabled`.

1 Install Apache HTTP Server and mod_jk.

- For information on installing Apache HTTP Server, see <http://httpd.apache.org/docs/2.2/install.html>.
- For information on installing mod_jk, see http://tomcat.apache.org/connectors-doc/webserver_howto/apache.html.

2 Configure the following files:

- `apache2/conf/httpd.conf`, the main Apache configuration file
- `apache2/conf/workers.properties` or `domain-dir/config/glassfish-jk.properties` (to use non-default values of attributes described at <http://tomcat.apache.org/tomcat-5.5-doc/config/ajp.html>)

If you use both the `workers.properties` file and the `glassfish-jk.properties` file, the file referenced by `httpd.conf` first takes precedence.

3 Start Apache HTTP Server (`httpd`).

4 Start GlassFish Server with at least one web application deployed.

In order for the `mod_jk`-enabled network listener to start listening for requests, the web container must be started. Normally, this is achieved by deploying a web application.

5 Create an HTTP listener by using the `create-http-listener(1)` subcommand.

Use the following format:

```
asadmin> create-http-listener --listenerport 8009
--listeneraddress 0.0.0.0 --defaultvs server listener-name
```

where *listener-name* is the name of the new listener.

6 Enable `mod_jk` by using the `set(1)` subcommand.

```
asadmin> create-network-listener --protocol http-listener-1 \
--listenerport 8009 --jkenabled true jk-connector
```

```
asadmin> set server-config.network-config.network-listeners.network-listener.\
jk-connector.jk-configuration-file=domain-dir/config/glassfish-jk.properties
```

7 If you are using the `glassfish-jk.properties` file and not referencing it in the `httpd.conf` file, point to the properties file by using the `create-jvm-options(1)` subcommand.

Use the following format:

```
asadmin> create-jvm-options -Dcom.sun.enterprise.web.connector.enableJK.propertyFile=
domain-dir/config/glassfish-jk.properties
```

8 To apply your changes, restart GlassFish Server.

See “[To Restart a Domain](#)” on page 89.

Example 6-4 httpd.conf File for mod_jk

This example shows an httpd.conf file that is set for mod_jk. In this example, mod_jk used as a simple pass-through.

```
LoadModule jk_module /usr/lib/httpd/modules/mod_jk.so
JkWorkersFile /etc/httpd/conf/worker.properties
# Where to put jk logs
JkLogFile /var/log/httpd/mod_jk.log
# Set the jk log level [debug/error/info]
JkLogLevel debug
# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
# JkOptions indicate to send SSL KEY SIZE,
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories
# JkRequestLogFormat set the request format
JkRequestLogFormat "%w %V %T"
# Send all jsp requests to GlassFish
JkMount /*.jsp worker1
# Send all glassfish-test requests to GlassFish
JkMount /glassfish-test/* worker1
```

Example 6-5 workers.properties File for mod_jk

This example shows a workers.properties or glassfish-jk.properties file that is set for mod_jk. This workers.properties file is referenced in Example [Example 6-4](#).

```
# Define 1 real worker using ajp13
worker.list=worker1
# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=localhost
worker.worker1.port=8009
```

See Also For more information on Apache, see <http://httpd.apache.org/>.

For more information on Apache Tomcat Connector, see <http://tomcat.apache.org/connectors-doc/index.html>.

▼ To Load Balance Using mod_jk and GlassFish Server

Load balancing is the process of dividing the amount of work that a computer has to do between two or more computers so that more work gets done in the same amount of time. Load balancing can be configured with or without security.

In order to support stickiness, the Apache mod_jk load balancer relies on a jvmRoute system property that is included in any JSESSIONID received by the load balancer. This means that every GlassFish Server instance that is front-ended by the Apache load balancer must be configured with a unique jvmRoute system property.

1 On each of the instances, perform the steps in “To Enable mod_jk” on page 114.

If your instances run on the same machine, you must choose different JK ports. The ports must match `worker.worker*.port` in your `workers.properties` file. See the properties file in [Example 6–5](#).

2 On each of the instances, create the `jvmRoute` system property of GlassFish Server by using the `create-jvm-options(1)` subcommand.

Use the following format:

```
asadmin> create-jvm-options "-DjvmRoute=/instance-worker-name/"
```

where `instance-worker-name` is the name of the worker that you defined to represent the instance in the `workers.properties` file.

3 To apply your changes, restart Apache HTTP Server and GlassFish Server.**Example 6–6** `httpd.conf` File for Load Balancing

This example shows an `httpd.conf` file that is set for load balancing.

```
LoadModule jk_module /usr/lib/httpd/modules/mod_jk.so
JkWorkersFile /etc/httpd/conf/worker.properties
# Where to put jk logs
JkLogFile /var/log/httpd/mod_jk.log
# Set the jk log level [debug/error/info]
JkLogLevel debug
# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
# JkOptions indicate to send SSL KEY SIZE,
JkOptions +ForwardKeySize +ForwardURICompact -ForwardDirectories
# JkRequestLogFormat set the request format
JkRequestLogFormat "%w %V %T"
# Send all jsp requests to GlassFish
JkMount /*.jsp worker1
# Send all glassfish-test requests to GlassFish
JkMount /glassfish-test/* loadbalancer
```

Example 6–7 `workers.properties` File for Load Balancing

This example shows a `workers.properties` or `glassfish-jk.properties` file that is set for load balancing. The `worker.worker*.port` should match with JK ports you created.

```
worker.list=worker1,worker2,loadbalancer
worker.worker1.type=ajp13
worker.worker1.host=localhost
worker.worker1.port=8009
worker.worker1.lbfactor=1
worker.worker1.socket_keepalive=1
worker.worker1.socket_timeout=300
worker.worker2.type=ajp13
worker.worker2.host=localhost
```

```
worker.worker2.port=8010
worker.worker2.lbfactor=1
worker.worker2.socket_keepalive=1
worker.worker2.socket_timeout=300
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=worker1,worker2
```

▼ To Enable SSL Between the mod_jk Load Balancer and the Browser

To activate security for mod_jk on GlassFish Server, you must first generate a Secure Socket Layer (SSL) self-signed certificate on the Apache HTTP Server with the mod_ssl module. The tasks include generating a private key, a Certificate Signing Request (CSR), a self-signed certificate, and configuring SSL-enabled virtual hosts.

Before You Begin The mod_jk connector must be enabled.

1 Generate the private key as follows:

```
openssl genrsa -des3 -rand file1:file2:file3:file4:file5 -out server.key 1024
```

where file1:file2: and so on represents the random compressed files.

2 Remove the pass-phrase from the key as follows:

```
openssl rsa -in server.key -out server.pem
```

3 Generate the CSR as follows:

```
openssl req -new -key server.pem -out server.csr
```

Enter the information you are prompted for.

4 Generate a temporary certificate as follows:

```
openssl x509 -req -days 60 -in server.csr -signkey server.pem -out server.crt
```

This temporary certificate is good for 60 days.

5 Create the http-ssl.conf file under the /etc/apache2/conf.d directory.

6 In the http-ssl.conf file, add one of the following redirects:

- Redirect a web application, for example, JkMount /hello/* worker1.
- Redirect all requests, for example, JkMount /* worker1.

```
# Send all jsp requests to GlassFish
JkMount /*.jsp worker1
# Send all glassfish-test requests to GlassFish
JkMount /glassfish-test/* loadbalancer
```

Example 6–8 http-ssl.conf File for mod_jk Security

A basic SSL-enabled virtual host will appear in the http-ssl.conf file. In this example, all requests are redirected.

```
Listen 443
<VirtualHost _default_:443>
SSLEngine on
SSLCipherSuite ALL:!ADH:!EXP56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLCertificateFile "/etc/apache2/2.2/server.crt"
SSLCertificateKeyFile "/etc/apache2/2.2/server.pem"
JkMount /* worker1
</VirtualHost>
```

▼ To Enable SSL Between the mod_jk Load Balancer and GlassFish Server

Before You Begin The self-signed certificate must be configured.

1 Perform the steps in [“To Enable mod_jk” on page 114](#).

2 Start another GlassFish Server with at least one web application deployed.

In order for the mod_jk-enabled network listener to start listening for requests, the web container must be started. Normally, this is achieved by deploying a web application.

3 Follow instructions from [“To Configure an HTTP Listener for SSL” on page 274 on the mod_jk connector](#).

Use the following format:

```
asadmin> create-ssl --type http-listener --certname sampleCert new-listener
```

4 Add the following directives in the httpd.conf file under the /etc/apache2/conf.d directory:

```
# Should mod_jk send SSL information (default is On)
JkExtractSSL On
# What is the indicator for SSL (default is HTTPS)
JkHTTPSIndicator HTTPS
# What is the indicator for SSL session (default is SSL_SESSION_ID)
JkSESSIONIndicator SSL_SESSION_ID
# What is the indicator for client SSL cipher suit (default is SSL_CIPHER)
JkCIPHERIndicator SSL_CIPHER
# What is the indicator for the client SSL certificated? (default is SSL_CLIENT_CERT)
JkCERTSIndicator SSL_CLIENT_CERT
```

5 To apply your changes, restart Apache HTTP Server and GlassFish Server.