

## Weighted Round Robin

The load balancer enables improved distribution of HTTP requests. The administrator can use an attribute called 'weight' to specify how requests will be proportionately routed to an instance. For example, suppose a cluster has two instances, and the administrator has assigned a weight of 100 to instance x and a weight of 400 to instance y. Now, for every 100 requests, 20 will go to instance x and 80 will go to instance y.

## User-Defined Load Balancing

GlassFish Server enables administrators to define custom policies for distributing HTTP requests. A custom policy defines the load balancing algorithm that the Loadbalancer Plug-In must use. In other words, an administrator can define which instance will handle an HTTP request. To use this feature, the administrator needs to develop a shared library, which must implement an interface called `loadbalancer.h`. The shared library can, for example, be used to evaluate the headers of incoming requests provided to it and in accordance to some criteria, select the instance that can serve the request. This shared library would be loaded by the load balancer.

The `loadbalancer.h` interface is available in `webserver-install-dir/glassfish-lbplugin/lib/install/templates`. A sample implementation, named `roundrobin.c`, is provided in this same directory. Administrators can use `roundrobin.c` as a template to build custom shared libraries.

**To Build the Sample User-Defined Load Balancing Algorithm on Solaris** Build the `roundrobin.c` sample on Solaris using one of the following compilers:

- The `gcc` compiler.

Enter the following command on a single line.

```
% gcc -m64 -shared
-Iwebserver-install-dir/glassfish-lbplugin/lib/install/templates roundrobin.c
-o roundrobin.so
```

If you get relocation errors, add the `-fPIC` option after the `-shared` option.

- The `cc` compiler, version 4.0.4 or later

Enter the following command on a single line.

```
% cc -m64 -G -Iwebserver-install-dir/glassfish-lbplugin/lib/install/templates
roundrobin.c -o roundrobin.so
```

If you get relocation errors, add the `-xcode=pic32` option after the `-G` option.

**To Build the Sample User-Defined Load Balancing Algorithm on Windows** Build the `roundrobin.c` sample on Windows using MSVC++ 2008 Express Edition, available at <http://www.microsoft.com/express/download/>. Before compiling the sample, ensure that the system environment variables related to MSVC are set:

- `INCLUDE=msvc-install-dir/VC/include`
- `LIB=msvc-install-dir/VC/lib;microsoft-sdk-lib-dir`

Additionally, ensure that the `PATH` system environment variable includes these directories:

- `msvc-install-dir/VC/bin`
- `msvc-install-dir/Common7/IDE`

To compile the sample, enter the following command on a single line.

```
D:\> cl /D_WINDOWS  
/Iwebserver-install-dir/glassfish-lbplugin/lib/install/templates /LD roundrobin.c
```

## Setting Up HTTP Load Balancing

This section describes how to set up load balancing for GlassFish Server.

The following topics are addressed here:

- [Prerequisites for Setting Up HTTP Load Balancing](#)
- [Procedure for Setting Up HTTP Load Balancing](#)
- [HTTP Load Balancer Deployments](#)

### Prerequisites for Setting Up HTTP Load Balancing

Before configuring your load balancer, you must:

- Install a supported web server and configure it. For more information on configuring a supported web server, see [Configuring Web Servers for HTTP Load Balancing](#).
- Install the Loadbalancer Plug-In.  
For complete instructions on installing the Loadbalancer Plug-In, see [Configuring Web Servers for HTTP Load Balancing](#).
- Create GlassFish Server clusters or server instances to participate in load balancing.
- Deploy applications to these clusters or instances.

### Procedure for Setting Up HTTP Load Balancing

There are several `asadmin` subcommands that enable you to set up and use HTTP load balancing in your GlassFish Server and web server environment.

#### To Set Up Load Balancing Using the `asadmin` Tool

Most of the required load balancing configuration settings can be performed in a single step with the `asadmin create-http-lb` subcommand, as described in this procedure. After performing this procedure, or if you choose to not perform all configuration steps with one subcommand, you can set individual load balancing options using several other subcommands. Information about configuring individual load balancing options is provided after this procedure, in [Configuring the HTTP Load Balancer](#).

1. Create and configure a load balancer.

Use the `asadmin create-http-lb` subcommand.

For the purposes of this procedure, the `create-http-lb` subcommand can be used to perform the following load balancing tasks. Other options are available but are not included in this example. For more information, see `create-http-lb`.

- Specify a target cluster or standalone server instance for the load balancer to manage (`--target`). This task can also be performed at a later time with the `create-http-lb-ref` subcommand.