

# Starting the Client Process

The LightWave Client process is a *server* application that runs on NonStop systems. It runs as either a Pathway serverclass or a standalone Guardian server process. The advantage of the Pathway option is Pathway's server process management and load-balancing features. In either case, LightWave Client supports a number of startup *command-line-options* that control various aspects of its operation. The same options are supported in both Pathway and standalone configurations, specified as server PARAMs or command-line-options, respectively. The object file name for LightWave Client is "CLIENT" and is located in the installation subvolume.

## Starting LightWave Client as a Standalone Server Process

LightWave Client may be started by running the CLIENT program from TACL.

```
tacl> run client / run-options / command-line-options
```

## Examples

The example starts the CLIENT process as \$LWC in CPU 0, supporting the service "MyService" located in filesystem \$DATA.LWFS.

```
tacl> run client / name $lwc, cpu 0 / --api MyService --base-url http://api.example.com --log $zhome info --standalone
```

## More Information

For complete information about *run-options* and *command-line-options* see the [CLIENT](#) section in [Command Line Reference](#).

## Starting LightWave Client as a Pathway Server



### Tip

A sample LightWave Client startup macro, ZSTARTPW, is included with the release. Before using for the first time, copy or rename the file to a new name (e.g. FUP DUP ZSTARTPW, STARTPW) so it won't be overwritten by future releases. The sample macro should be suitable for most installations, but it will be necessary to modify certain parameters for your API and environment. To use the macro: RUN STARTPW

The LightWave Client Process may be configured as a Pathway server, which allows you to take advantage of Pathway's process management and load-balancing features. The configuration options are the same as those for standalone process configuration, however in the Pathway environment individual options are supplied as server PARAMs. The following is a sample pathway server configuration. Note that the server settings here (createdelay, deletedelay, maxservers, etc.) are merely examples, not requirements or recommendations.

```
reset server
set server cpus 0:1
set server createdelay 0 secs
set server deletedelay 60 secs
set server highpin on
set server linkdepth 1
set server maxservers 6
set server maxlinks 20
set server numstatic 0
set server program CLIENT
set server tmf on
set server debug off
set server param api "MyService"
set server param base-url "http://api.example.com"
set server param log "$0 info text"
add server MY-SERVICE
```

You may also provide the options in a command file by setting the SERVER STARTUP value to @<command-file>. The following is a sample pathway configuration using this technique.

```
reset server
set server cpus 0:1
set server createdelay 0 secs
set server deletedelay 60 secs
set server highpin on
set server linkdepth 1
set server maxservers 6
set server maxlinks 20
set server numstatic 0
set server program CLIENT
set server tmf on
set server debug off
set server startup @config
add server MY-SERVICE
```