

CLIENT

CLIENT is the LightWave Client process, which is responsible for relaying messages between a client application and a web service based on the contents of the API Definition. The process is started by running the CLIENT program from TACL or by configuring the program as a Pathway Server Class. This section describes how to start the program and available program options.

- [Starting CLIENT as Standalone Process](#)
- [Configuring CLIENT as a Pathway Server Class](#)
 - [Configuration Using the Server Class STARTUP Attribute](#)
 - [Configuration using Server Class PARAMS](#)
- [CLIENT Program Options](#)
- [Remarks](#)
- [Examples](#)
 - [Standalone Process](#)
 - [Pathway Server Class Using STARTUP Attribute](#)
 - [Pathway Server Class Using PARAMS](#)

Starting CLIENT as Standalone Process

The CLIENT process may be started by running the CLIENT program from TACL.

```
tacl > run CLIENT / run-options / program-options
```

The "--api" and "--base-url" command-line-options are required. All others are optional. The run-options are the standard TACL run options. Note that the process does not optn the IN or OUT file. You should be logged-on as a user with sufficient privileges to access the system resources that the process requires.

Configuring CLIENT as a Pathway Server Class

The CLIENT process may be configured as a Pathway Server Class. This is the preferred method, as it allows CLIENT processes to be created and deleted to meet application demand. When configuring a Server Class, program options may be specified with the STARTUP attribute or specified as PARAMS.

Configuration Using the Server Class STARTUP Attribute

Program options may be supplied directly in the STARTUP string or entered into an EDIT file and supplied using a Command File.

Server Class configuration

```
reset server
set server program      client
set server startup      "--api myapi --base-url http://api.example.com"
```

Or

Server Class configuration

```
reset server
set server program      client
set server startup      "@cmdfile"
```

```
EDIT file cmdfile contents:
--api myapi
--base-url https://api.example.com
--log logfile info
```

Using the STARTUP attribute with a command file allows the program options to be modified without re-configuring the Server Class. Note that changes to the command file take effect when a new CLIENT process is started and have no effect on processes that are already running.

Configuration using Server Class PARAMS

Program options may be specified as individual PARAMs. When options are specified as PARAMs, do not include the leading '--' characters in the PARAM name which would make the name invalid. The PARAM values should be enclosed in quotes:

Server Class configuration

```
set server param api           "myapi"  
set server param base-url     "http://api.example.com"
```

Some program options, such as *cert-no-verify*, do not require a value. The presence of these options on the command line activates the feature. Because Pathway PARAMs require a value, when specifying these option as a PARAM, specify any value for the PARAM to activate the feature. The value itself is ignored. For example:

Server Class configuration

```
set server param cert-no-verify "1"  
set server param cert-no-verify "true"
```

Because the PARAM value is ignored, both of these examples will activate the *cert-no-verify* feature.

CLIENT Program Options

@<command-file>

Reads command line options from <command-file>. Options specified on the command line override any duplicates specified in the file. At most, one '@' option may be used. The file itself cannot contain an '@' option (i.e., no nesting).

--api <file-name>

The name of an API definition file. This option is required. The API definition must be exported from the LightWave Client filesystem into an API definition file through the LightWave Client Console or by using the CUTILITY -- export-api command. Changes to API definition can be incorporated while CLIENT is running (i.e., without restarting) if the --monitor api option is used.

--api-param-<param-name> <param-value>

The API parameter value for the parameter <param-name>. This option should be specified for each parameter defined in the API definition. See [Working with API Parameters](#).

--auth <file-name>

The name of an auth config file. For more information on auth configuration and request signing, see [Request Authentication and Signing](#).

--base-url <url>

The base URL of the target web service in the form *http[s]://host[:port]/[base-path]*. Note that the optional base-path will be concatenated with the API operation path to form the full URL. This option is required.

--blob-files [\$vol].subvol].file-name-prefix [userid,groupid | groupname,username] [security-string] [extents=<pri>,<sec>,<max>]

A pattern which specifies the file system location and file name prefix for output BLOB files and optionally, the user id and file security. The file name prefix is limited to 1 to 3 characters with the remaining 5 characters assigned by the CLIENT process. If the volume or subvolume portion of the pattern is omitted, the process default volume and subvolume are used. If the option is omitted, the default pattern is "\$current-vol.current-subvol.BLB" and the userid and file security are that of the CLIENT process. Note that client applications are responsible for disposing of the BLOB files once they have been processed.

--ca-root-certs <file-name>

The name of the file containing the certificates of all trusted root certificate authorities. If omitted, this value defaults the CAROOT file that is provided with the LightWave Client software.

--ca-local-certs <file-name>

The name of a file containing the certificates of local trusted certificate authorities. If omitted, no local CA certificates are loaded.

--cert-no-verify

The presence of this option indicates that the CLIENT process should not validate the server certificate for common name, expiration date, or issuer when a secure connection is established. Note that this option should only be used in a development/test environment where the server may not necessarily pass all of the verification criteria. It should not be used in a production environment.

--client-cert <certificate-file> [{ <pass phrase> | +<credentials-file> }]

The client certificate to use for the secure connection. Specify the name of the file and, if required, the associated pass phrase required to access the certificate. Specify the pass phrase, in plain text, or the name of an existing LightWave Credentials file containing the encrypted pass phrase.

--default-encoding <encoding-name>

Specifies the default encoding to use for character string conversions for the API. Note that encoding settings applied to API method or data type definitions will override this setting. The <encoding-name> must be one of the names listed in [Character Encoding Names](#). If omitted, the default encoding is ISO-8859-1.

--diag-log <config-spec> | +<diag-log-config-file> | <subvolume-name>

Enables diagnostic logging and specifies the subvolume where the logs are stored, the location of a log config file, or a string consisting of diagnostic log configuration options. Log files are named using the format DLnnnnnn where nnnnnn is a sequence number. The logs may be viewed using command line tools or from the LightWave Client Console if it is installed. See [Diagnostic Log Configuration](#) for information about diagnostic logging configuration files and config-spec options. Note that the <subvolume-name> option is available for compatibility with release prior to 1.0.5. Using a config-file or config-spec is recommended.

--disable-sensitive-data-masking

When present, the [sensitive data masking](#) feature is disabled and fields marked as sensitive will be displayed in HTTP and diagnostic logs. This option should only be used during application development when sensitive data is not contained in message payloads.

--http-credentials { <userid>:<password> | +<credentials-file> } [pre-auth]

The credentials required for the HTTP connection to the Web service host when HTTP Basic or Digest authentication is required. Specify either your userid and password (in plain text) or the name of an existing credentials file containing the encrypted userid and password. See [Using Credentials Files](#) for information about creating credentials files. Supplying "pre-auth" indicates that pre-authentication should be used. This causes Basic authentication credentials to be sent with every request, without waiting for an HTTP 401 response. Use of this option can improve performance on connections that use HTTP Basic authentication but has serious security implications for non-HTTPS (TLS) connections. The "pre-auth" option should not be used unless the security implications are fully understood.

--http-proxy-host <address[:port]>

The host name or IP address and port of an HTTP proxy that should be used for HTTP/HTTPS connections. If omitted, no proxy is used. If the port value is omitted, port 80 is used.

--http-proxy-credentials { <userid>:<password> | +<credentials-file> }

The credentials required for the HTTP proxy. Specify either your userid and password (in plain text) or the name of an existing credentials file containing the encrypted userid and password. See [Using Credentials Files](#) for information about creating credentials files.

--http-request-timeout <milliseconds> [!]

The number of milliseconds to wait for a web service request/response exchange to complete. If omitted, the default value of 60 seconds (60000 milliseconds) is used. If '!' is specified, this value will override any value set by the client application in the rq_timeout field of the LightWave request header. If '!' is not specified, and the client application specifies a timeout value in the rq_timeout field, the value in the rq_timeout field is used.

--license <file-name>

The name of an existing edit file containing the LightWave Client product license. If this option is omitted, the license file is located according to [Product Licensing](#) rules.

--log [{ <destination> | * } [level [format]] | +<log-config-file>]

Specifies the process log location, the level, and the log event format, or the location of a log configuration file. The destination value may be a process name, a file name, or the asterisk (*) character. If the asterisk is used then the log output is directed to the home term of the process. The level value may be "error", "warning", "info", or "debug" and controls the type of information that is output to the log destination. The "error" level produces the least output while the "debug" level produces the most output. The format value may be "text" indicating that the log events should be output as text strings or "event" indicating that the log events should be output in EMS event format. If omitted, the default is "-log * info text". See [Using Configuration Files](#) for information about logging configuration files.

--monitor <option>[:<interval>] [<option>[:<interval>]] ...

Enables file monitoring and specifies the monitoring interval. If the interval is omitted, the default value is 15 seconds. The following files may be monitored: api, log, diag-log. See [Using Configuration Files](#) for information about monitoring log and diag-log configuration files.

--standalone

The presence of this options causes the process to ignore close messages. In order to function properly in the Pathway environment, the CLIENT process, by default, matches open and close messages and terminates when all clients have closed the process. This option can be used to prevent the process from terminating when it is run as a standalone process.

--string-padding

Specifies a string padding value that will override the string padding setting in the API definition. The value may be 'zeros', 'spaces', or an integer in the range 0 - 255.

--tcpip-bind-addr <ipv4-address>

Specifies an IPv4 address to bind to for TCP/IP connections. This option may be used to specify the IP address from which connections will originate when a TCPIP provider is configured with multiple IP addresses. If omitted, the default IP address for the TCPIP provider is used.

--tcpip-process <process name>

Specifies the name of the TCPIP process that the process should use. If omitted, the value of the =TCPIP^PROCESS^NAME MAP define is used if it exists, otherwise \$ZTCO is used.

Remarks

All command-line-option names and values are case-insensitive except where noted. If multiple occurrences of the same command line parameter are encountered, the setting of the last occurrence is used.

When the -log option format is set to event, EMS events will be sent to the output device with the following EMS Subsys ID:

Z-OWNER	"NUWAVE"
Z-NUMBER	4
Z-VERSION	Product major version

Examples

Standalone Process

Starting from TACL using options on the command line

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

Starting from TACL using a command file

```
tacl> run CLIENT / name $lwc, nowait, term $zhome / @cmdfile
```

```
EDIT file cmdfile contents:  
--standalone  
--api employee  
--base-url http://api.example.com  
--log $0 info
```

Specify values for API parameters username and api-key

```
tacl> run CLIENT / name $lwc, nowait, term $zhome / --standalone --api employee --base-url http://api.example.com --api-param-username johnsmith --api-param-api-key 35bddf603d7b4cef9fbaf1689c1cd49e
```

Monitor the API and log configuration files for changes

```
tacl> run CLIENT / name $lwc, nowait, term $zhome / --standalone --api employee --base-url http://api.example.com --log +logcfg --monitor api:30 log:15
```

Pathway Server Class Using STARTUP Attribute

Server Class configuration

== Settings for linkdepth, maxservers, maxlinks, etc are examples, not recommendations.

```
reset server
set server cpus          0:1
set server createdelay   1 secs
set server deletedelay   120 secs
set server highpin       on
set server linkdepth     10
set server maxservers    10
set server maxlinks      10
set server numstatic     0
set server program       client
set server tmf           off
set server debug         off
set server startup       "@cmdfile"
add server example
```

EDIT file cmdfile contents:

```
--api employee
--base-url http://api.example.com
--log +logcfg
--monitor api:30 log:15
```

Pathway Server Class Using PARAMs

Server Class configuration

== Settings for linkdepth, maxservers, maxlinks, etc are examples, not recommendations.

```
reset server
set server cpus          0:1
set server createdelay   0 secs
set server deletedelay   15 secs
set server highpin       on
set server linkdepth     5
set server maxservers    1
set server maxlinks      20
set server numstatic     0
set server program       client
set server tmf           off
set server debug         off
set server param         api "employee"
set server param         base-url "https://api.example.com"
set server param         diag-log "+logconf"
set server param         log "+logconf"
set server param         monitor "api:5 diag-log:5 log:5"
set server param         tcpip-process "$ztc0"
add server example
```