

# Features

---

 [ritlabs.com/en/products/tinyweb/features.php](https://ritlabs.com/en/products/tinyweb/features.php)

## What does TinyWeb do

### Handles http-requests!

Path to home directory is taken from command line as a first parameter. The second parameter is bind-port and the third is bind-address, both are optional, don't use them if unsure.

The default directory root file name is `index.html`. If `index.html` file is not found in a directory, TinyWEB then looks for `index.htm`. If none of these two files are found, a file with name `index` and extension from `PATHEXT` environment variable is executed as a CGI from `cgi-bin` directory. If you have no such files in `cgi-bin`, make sure that `index.html` or `index.htm` exist in the directory first command line parameter points to. Please refer to `error_log` in case of any troubles.

The only `GET`, `HEAD` and `POST` commands are handled.

### Executes CGI applications!

The following environment variables are passed to application:

- `PATH_INFO`
- `PATH_TRANSLATED`
- `REMOTE_HOST`
- `REMOTE_ADDR`
- `GATEWAY_INTERFACE`
- `SCRIPT_NAME`
- `REQUEST_METHOD`
- `HTTP_ACCEPT`
- `HTTP_ACCEPT_CHARSET`
- `HTTP_ACCEPT_ENCODING`
- `HTTP_ACCEPT_LANGUAGE`
- `HTTP_FROM`
- `HTTP_HOST`
- `HTTP_REFERER`
- `HTTP_USER_AGENT`
- `QUERY_STRING`
- `SERVER_SOFTWARE`
- `SERVER_NAME`
- `SERVER_PROTOCOL`
- `SERVER_PORT`
- `CONTENT_TYPE`
- `CONTENT_LENGTH`

- USER\_NAME
- USER\_PASSWORD
- AUTH\_TYPE

To learn more about CGI, please search for "Common Gateway Interface".

CGI interface was tested with ActivePerl for NT, which is available at <http://www.activestate.com/>

CGI application are being executed from `/cgi-bin/` directory only. All files requested from `/cgi-bin/` directory will be treated as CGI and executed, so don't put non-CGI files into `/cgi-bin/` directory.

TinyWeb handles "Location" and "Status" CGI script output directives. "Location" may only contain an URL, local file redirections are not supported.

The samples to test CGI are provided in `cgitest.zip` package within TinyWeb archive.

Note that TinyWeb extracts content-types, associated to file extensions, and script interpreters (e.g. Perl) from Windows Registry, so make sure that `.html`, `.htm`, `.pl` are registered file types. It means that I recommend to use `.pl` as an extension for CGI-scripts written in Perl.

The information is being taken from registry (to TinyWeb's cache) on TinyWeb's startup, so if you made some content-type or file-association-related changes of registry, you must reload TinyWeb in order to take effect.

## Keeps logs!

TinyWeb keeps Apache-compatible [The Common Logfile Format](#) log files: `access_log`, `agent_log`, `referer_log` and `error_log`, which can be analyzed by any compatible log analyzer.

Logs are kept in a directory from that TinyWeb was started. Make sure that "Start In" directory of TinyWeb properties is set properly.

Please note that TinyWeb doesn't support [W3C's Extended Log File Format](#).

## Allows using CGI instead of index.html

As mentioned below, if an URI with empty name part specified (only path part is specified) and no `index.html` or `index.htm` found on this path, TinyWeb can run CGI application. Examples when only path part is specified are `http://www.rtitlabs.com/` or `http://www.rtitlabs.com/tinyweb/`. In that case TinyWEB first looks for `index.html` or `index.htm` and, if none of them found, runs a file with name `index` and extension from `PATHEXT` environment variable. For example, if `PATHEXT` is set to `.pl` and `/mydir/` is requested, `/cgi-bin/mydir/index.pl` will be run. TinyWeb caches these CGI handler application pathnames, so if you had `/cgi-bin/mydir/index.pl` for `/mydir/` and then removed that `index.pl` and put `index.exe` instead, you should reload TinyWeb.

## What does TinyWeb not support

### Administration and setup

- Administer server from browser
- Graphical wizard-based automated setup routine
- Non-browser graphical-based setup
- GUI-based maintenance
- Remote maintenance

- Real-time performance measurement tools

## Logging

- Logging with Event Log
- W3C extended log format
- Log to ODBC database
- Log files can be automatically cycled or archived
- Normal (hit) log entries can be customized
- Possiblilty write to multiple logs
- Generating non-hit log entries and comments
- CGI scripts creating their own log entries
- Performance measurment logs
- Tracking individual users in log

## Protocol Support and Includes

- Selecting documents based on Accept header
- Selecting documents based on User-Agent header
- Including based on HTML comments
- Including based on request headers
- Forced includes
- Automatic include if any HTTP headers in responses
- Built-in scripting language
- Built-in image-map handling
- Support of HTTP/1.1 PUT
- Understanding of full URIs in HTTP/1.1 requests
- Non-supported methods to invoke a scrip
- SNMP agent
- Support of Netscape Server API
- Support of Microsoft ISAPI

## Security

- Support of PCT
- Prohibiting access by domain name
- Prohibiting access by IP address
- Configurable user groups, not just a single user list
- Hiding part of a document based on security rules
- Security rules based on URLs
- Integrated certificate server

## Searching and Indexing

- Internal search engine
- Scriptable and programable searches
- Automatic index updates when changes are made
- Limiting queries by directory
- Highlighting hits from searches
- Return document properties in query results

## **Other Features**

- Script or action based on output media type
- Automatic directory tree
- User directories
- Direct (non-CGI) link to a DBMS
- User interaction tools
- NCSA-server-side imagemaps
- Non-blocking DNS
- Act as caching HTTP proxy server