

## Apache Module mod\_cgi

|                           |                          |
|---------------------------|--------------------------|
| <b>Description:</b>       | Execution of CGI scripts |
| <b>Status:</b>            | Base                     |
| <b>Module Identifier:</b> | cgi_module               |
| <b>Source File:</b>       | mod_cgi.c                |

### Summary

Any file that has the mime type `application/x-httpd-cgi` or handler `cgi-script` (Apache 1.1 or later) will be treated as a CGI script, and run by the server, with its output being returned to the client. Files acquire this type either by having a name containing an extension defined by the `AddType` directive, or by being in a `ScriptAlias` directory.

When the server invokes a CGI script, it will add a variable called `DOCUMENT_ROOT` to the environment. This variable will contain the value of the `DocumentRoot` configuration variable.

For an introduction to using CGI scripts with Apache, see our tutorial on [Dynamic Content With CGI](#)<sup>1</sup>.

When using a multi-threaded MPM under unix, the module `mod_cgid` should be used in place of this module. At the user level, the two modules are essentially identical.

### Topics

|                                |   |
|--------------------------------|---|
| CGI Environment variables..... | 1 |
| CGI Debugging .....            | 2 |
| URI References .....           | 3 |

### Directives

|                       |   |
|-----------------------|---|
| ScriptLog .....       | 2 |
| ScriptLogBuffer ..... | 3 |
| ScriptLogLength.....  | 3 |

### See also

- [AcceptPathInfo](#)
- [Options](#)
- [ScriptAlias](#)
- [AddHandler](#)
- [Running CGI programs under different user IDs](#)<sup>2</sup>
- [CGI Specification](#)<sup>3</sup>

## CGI Environment variables

---

The server will set the CGI environment variables as described in the CGI specification<sup>3</sup>, with the following provisions:

### **PATH\_INFO**

This will not be available if the `AcceptPathInfo` directive is explicitly set to `off`. The default behavior, if `AcceptPathInfo` is not given, is that `mod_cgi` will accept path info (trailing `/more/path/info` following the script filename in the URI), while the core server will return a 404 NOT FOUND error for requests with additional path info. Omitting the `AcceptPathInfo` directive has the same effect as setting it `On` for `mod_cgi` requests.

### **REMOTE\_HOST**

---

## Apache Module mod\_cgi

---

This will only be set if `HostnameLookups` is set to `on` (it is off by default), and if a reverse DNS lookup of the accessing host's address indeed finds a host name.

### REMOTE\_IDENT

This will only be set if `IdentityCheck` is set to `on` and the accessing host supports the ident protocol. Note that the contents of this variable cannot be relied upon because it can easily be faked, and if there is a proxy between the client and the server, it is usually totally useless.

### REMOTE\_USER

This will only be set if the CGI script is subject to authentication.

## CGI Debugging

---

Debugging CGI scripts has traditionally been difficult, mainly because it has not been possible to study the output (standard output and error) for scripts which are failing to run properly. These directives, included in Apache 1.2 and later, provide more detailed logging of errors when they occur.

### CGI Logfile Format

When configured, the CGI error log logs any CGI which does not execute properly. Each CGI script which fails to operate causes several lines of information to be logged. The first two lines are always of the format:

```
%% [time] request-line
%% HTTP-status CGI-script-filename
```

If the error is that CGI script cannot be run, the log file will contain an extra two lines:

```
%%error
error-message
```

Alternatively, if the error is the result of the script returning incorrect header information (often due to a bug in the script), the following information is logged:

```
%request
All HTTP request headers received
POST or PUT entity (if any)
%response
All headers output by the CGI script
%stdout
CGI standard output
%stderr
CGI standard error
```

(The `%stdout` and `%stderr` parts may be missing if the script did not output anything on standard output or standard error).

## ScriptLog Directive

---

|                     |  |
|---------------------|--|
| <b>Description:</b> | Location of the CGI script error logfile |
| <b>Syntax:</b>      | <code>ScriptLog file-path</code>         |
| <b>Context:</b>     | server config, virtual host              |
| <b>Status:</b>      | Base                                     |

---

## Apache Module mod\_cgi

---

|                                  |
|----------------------------------|
| <b>Module:</b> mod_cgi, mod_cgid |
|----------------------------------|

The `ScriptLog` directive sets the CGI script error logfile. If no `ScriptLog` is given, no error log is created. If given, any CGI errors are logged into the filename given as argument. If this is a relative file or path it is taken relative to the `ServerRoot`.

### Example

```
ScriptLog logs/cgi_log
```

This log will be opened as the user the child processes run as, *i.e.* the user specified in the main `User` directive. This means that either the directory the script log is in needs to be writable by that user or the file needs to be manually created and set to be writable by that user. If you place the script log in your main logs directory, do **NOT** change the directory permissions to make it writable by the user the child processes run as.

Note that script logging is meant to be a debugging feature when writing CGI scripts, and is not meant to be activated continuously on running servers. It is not optimized for speed or efficiency, and may have security problems if used in a manner other than that for which it was designed.

## ScriptLogBuffer Directive

---

|                     |   |
|---------------------|---|
| <b>Description:</b> | Maximum amount of PUT or POST requests that will be recorded in the scriptlog |
| <b>Syntax:</b>      | <code>ScriptLogBuffer bytes</code>  |
| <b>Default:</b>     | <code>ScriptLogBuffer 1024</code>   |
| <b>Context:</b>     | server config, virtual host   |
| <b>Status:</b>      | Base  |
| <b>Module:</b>      | mod_cgi, mod_cgid   |

The size of any PUT or POST entity body that is logged to the file is limited, to prevent the log file growing too big too quickly if large bodies are being received. By default, up to 1024 bytes are logged, but this can be changed with this directive.

## ScriptLogLength Directive

---

|                     |                                       |
|---------------------|---------------------------------------|
| <b>Description:</b> | Size limit of the CGI script logfile  |
| <b>Syntax:</b>      | <code>ScriptLogLength bytes</code>    |
| <b>Default:</b>     | <code>ScriptLogLength 10385760</code> |
| <b>Context:</b>     | server config, virtual host           |
| <b>Status:</b>      | Base                                  |
| <b>Module:</b>      | mod_cgi, mod_cgid                     |

`ScriptLogLength` can be used to limit the size of the CGI script logfile. Since the logfile logs a lot of information per CGI error (all request headers, all script output) it can grow to be a big file. To prevent problems due to unbounded growth, this directive can be used to set an maximum file-size for the CGI logfile. If the file exceeds this size, no more information will be written to it.

## URI References

---

## Apache Module mod\_cgi

---

- [1] <http://httpd.apache.org/docs-2.1/howto/cgi.html>
- [2] <http://httpd.apache.org/docs-2.1/suexec.html>
- [3] <http://hoohoo.ncsa.uiuc.edu/cgi/>