APPENDIX D

Environment Variables

IN THIS APPENDIX, I'll present the environment variables.

Standard Variables

The environment variables in Table D-1 aren't request-specific and are set for all requests.

Table D-1. Request-Independent Variables

Variable Name	Description
GATEWAY_INTERFACE	The revision of the CGI specification to which this server complies. The format is CGI/revision.
SERVER_NAME	The server's hostname, DNS alias, or IP address as it would appear in self-referencing URLs.
SERVER_SOFTWARE	The name and version of the information server software answering the request (and running the gateway). The format is name/version.

The environment variables in Table D-2 are specific to the request being fulfilled by Apache.

Table D-2. Request-Specific Variables

Variable Name	Description
AUTH_TYPE	If the server supports user authentication and the script is protected, this is the protocol-specific authentication method used to validate the user.
CONTENT_LENGTH	The length of the said content as given by the client.
CONTENT_TYPE	For queries that have attached information, such as HTTP POST and PUT, this is the content type of the data.
PATH_INFO	The extra path information, as given by the client. In other words, scripts can be accessed by their virtual pathname, followed by extra information at the end of this path. The extra information is sent as PATH_INFO. This information should be decoded by the server if it comes from a URL before it's passed to the CGI script.
PATH_TRANSLATED	The server provides a translated version of PATH_INFO, which takes the path and does any virtual-to-physical mapping to it.

(Continued)

Table D-2. Request-Specific Variables (Continued)

Variable Name	Description
QUERY_STRING	The information that follows the ? in the URL that referenced this script. This is the query information. It shouldn't be decoded in any fashion. This variable should always be set when there is query information, regardless of command-line decoding.
REMOTE_ADDR	The IP address of the remote host making the request.
REMOTE_HOST	The hostname making the request. If the server doesn't have this information, it should set REMOTE_ADDR and leave this unset. In Apache, this will only be set if HostnameLookups is set to on (it's off by default) and if a reverse DNS lookup of the accessing host's address indeed finds a hostname.
REMOTE_IDENT	If the HTTP server supports RFC 931 identification, then this variable will be set to the remote username retrieved from the server. Usage of this variable should be limited to logging only. In Apache, 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 can't be relied upon because it can easily be faked, and if there's a proxy between the client and the server, it's usually totally useless.
REMOTE_USER	If the server supports user authentication and the script is protected, this is the username they have authenticated as. In Apache, this will only be set if the CGI script is subject to authentication.
REQUEST_METHOD	The method with which the request was made. For HTTP, this is GET, HEAD, POST, and so on.
SCRIPT_NAME	A virtual path to the script being executed, which is used for self-referencing URLs.
SERVER_PORT	The port number to which the request was sent.
SERVER_PROTOCOL	The name and revision of the information protocol this request came in with. The format is protocol/revision.

Header Variables

In addition to these, the header lines received from the client, if any, are placed into the environment with the prefix HTTP_ followed by the header name in capital letters (see Table D-3). Any - characters in the header name are changed to _ characters. The server may exclude any headers that it has already processed, such as Authorization, Content-type, and Content-length. If necessary, the server may choose to exclude any or all of these headers if including them would exceed any system environment limits.

An example of this is the HTTP_ACCEPT variable that was defined in CGI/1.0. Another example is the header User-Agent.

Table D-3. Header Variables

Variable Name	Description
HTTP_ACCEPT	The MIME types that the client will accept, as given by HTTP headers. Other protocols may need to get this information from elsewhere. Each item in this list should be separated by commas as per the HTTP specification. The format is `type/subtype`, `type/subtype`.
HTTP_ACCEPT_CHARSET	A listing of character sets that can be processed by the client.
HTTP_ACCEPT_ENCODING	The coding types that can be processed by the client.
HTTP_ACCEPT_LANGUAGE	The languages that can be processed by the client or, rather, by the client's user.
HTTP_AUTHORIZATION	The data of an HTTP authentication.
HTTP_CACHE_CONTROL	Information regarding whether and how an object can be saved and/or whether a saved object is permitted to be returned by a cache.
HTTP_COOKIE	The cookie transmitted by the client, provided it exists.
HTTP_FORWARDED	The old variant of the `via` header.
HTTP_FROM	The e-mail address of the client or its user (only transmitted very rarely).
HTTP_HOST	The name of the Web server addressed by the client.
HTTP_PRAGMA	The old HTTP/1.0 variant of the `Cache-Control` header. Typically used to instruct a proxy cache to request the object again from the Web server concerned.
HTTP_REFERER	The URL of the page from which a link was traced.
HTTP_USER_AGENT	The browser the client is using to send the request. The general format is `software/version`, `library/version`.
HTTP_VIA	Information regarding proxy caches used for making the request.

Apache's Own Variables

Table D-4 describes the Apache variables.

Table D-4. Apache's Own Variables

Variable Name	Description
DOCUMENT_PATH_INFO	The additional path information that was passed to a document.
DOCUMENT_ROOT	The file path specified with the command `DocumentRoot`.
PATH	Corresponds to the shell variable of the same name. It contains the path specifications that were set in this location when Apache was started. The CGI scripts and SSI shell commands are also executed with this path.
REMOTE_PORT	The port used on the client side.

(Continued)

Table D-4. Apache's Own Variables (Continued)

Variable Name	Description
REQUEST_URI	The URL path of the file called. This is set by mod_rewrite.
SCRIPT_NAME	The URL path of the CGI script that was called.
SCRIPT_FILENAME	The absolute file path of the CGI script called. Just as most other SCRIPT variables, it's also set when a normal document is called.
SCRIPT_URI	The absolute URL (including the hostname) of the CGI script called. This is set by mod_rewrite.
SCRIPT_URL	The URL path of the CGI script that was called. This is comparable to SCRIPT_NAME and DOCUMENT_URL and is set by mod_rewrite.
SERVER_ADMIN	The e-mail address specified with ServerAdmin.

Variables Set by mod_include

In addition to the variables in the standard CGI environment, the variables in Table D-5 are available for the echo command, for if and elif, and to any program invoked by the document.

Table D-5. mod_include *Variables*

Name	Description
DATE_GMT	The current date in Greenwich Mean Time.
DATE_LOCAL	The current date in the local time zone.
DOCUMENT_NAME	The filename (excluding directories) of the document requested by the user.
DOCUMENT_URI	The (%-decoded) URL path of the document requested by the user. Note that in the case of nested include files, this isn't the URL for the current document.
LAST_MODIFIED	The last modification date of the document requested by the user.
USER_NAME	The name of the user who started Apache.

Special-Purpose Environment Variables

Interoperability problems have led to the introduction of mechanisms to modify the way Apache behaves when talking to particular clients. To make these mechanisms as flexible as possible, Apache lets you invoke them by defining environment variables (see Table D-6), typically with `BrowserMatch`, though `SetEnv` and `PassEnv` could also be used.

Table D-6. Special-Purpose Variables

Variable Name	Description
`downgrade-1.0`	This forces the request to be treated as a HTTP/1.0 request even if it claims to be in a later dialect.
`force-no-vary`	This causes any `Vary` fields to be removed from the response header before it's sent back to the client. Some clients don't interpret this field correctly; setting this variable can work around this problem. Setting this variable also implies `force-response-1.0`.
`force-response-1.0`	This forces an HTTP/1.0 response when set. It was originally implemented as a result of a problem with AOL's proxies. Some clients may not behave correctly when given an HTTP/1.1 response, and this can be used to interoperate with them.
`nokeepalive`	This disables `KeepAlive` when set. Because of problems with Netscape 2.*x* and `KeepAlive`, the following directive is normally used: `BrowserMatch Mozilla/2 nokeepalive`
`redirect-carefully`	This forces the server to be more careful when sending a redirect to the client. This is typically used when a client has a known problem handling redirects. This was originally implemented as a result of a problem with Microsoft's WebFolders software, which has a problem handling redirects on directory resources via `DAV` methods.