

Apache Module `mod_setenvif`

Description:	Allows the setting of environment variables based on characteristics of the request
Status:	Base
Module Identifier:	<code>setenvif_module</code>
Source File:	<code>mod_setenvif.c</code>

Summary

The `mod_setenvif` module allows you to set environment variables according to whether different aspects of the request match regular expressions you specify. These environment variables can be used by other parts of the server to make decisions about actions to be taken.

The directives are considered in the order they appear in the configuration files. So more complex sequences can be used, such as this example, which sets `netscape` if the browser is mozilla but not MSIE.

```
BrowserMatch ^Mozilla netscape
BrowserMatch MSIE !netscape
```

Topics

URI References	4
----------------------	---

Directives

<code>BrowserMatch</code>	1
<code>BrowserMatchNoCase</code>	2
<code>SetEnvIf</code>	2
<code>SetEnvIfNoCase</code>	4

See also

- Environment Variables in Apache¹

BrowserMatch Directive

Description:	Sets environment variables conditional on HTTP User-Agent
Syntax:	<code>BrowserMatch regex [!]env-variable[=value]</code> <code>[!]env-variable[=value] ...</code>
Context:	server config, virtual host, directory, <code>.htaccess</code>
Override:	<code>FileInfo</code>
Status:	Base
Module:	<code>mod_setenvif</code>

The `BrowserMatch` is a special cases of the `SetEnvIf` directive that sets environment variables conditional on the `User-Agent` HTTP request header. The following two lines have the same effect:

```
BrowserMatchNoCase Robot is_a_robot
SetEnvIfNoCase User-Agent Robot is_a_robot
```

Some additional examples:

 Apache Module mod_setenvif

```
BrowserMatch ^Mozilla forms jpeg=yes browser=netscape
BrowserMatch "^Mozilla/[2-3]" tables agif frames javascript
BrowserMatch MSIE !javascript
```

BrowserMatchNoCase Directive

Description:	Sets environment variables conditional on User-Agent without respect to case
Syntax:	BrowserMatchNoCase <i>regex</i> [!]env-variable[=value] [[!]env-variable[=value]] ...
Context:	server config, virtual host, directory, .htaccess
Override:	FileInfo
Status:	Base
Module:	mod_setenvif
Compatibility:	Apache 1.2 and above (in Apache 1.2 this directive was found in the now-obsolete mod_browser module)

The `BrowserMatchNoCase` directive is semantically identical to the `BrowserMatch` directive. However, it provides for case-insensitive matching. For example:

```
BrowserMatchNoCase mac platform=macintosh
BrowserMatchNoCase win platform=windows
```

The `BrowserMatch` and `BrowserMatchNoCase` directives are special cases of the `SetEnvIf` and `SetEnvIfNoCase` directives. The following two lines have the same effect:

```
BrowserMatchNoCase Robot is_a_robot
SetEnvIfNoCase User-Agent Robot is_a_robot
```

SetEnvIf Directive

Description:	Sets environment variables based on attributes of the request
Syntax:	SetEnvIf <i>attribute regex</i> [!]env-variable[=value] [[!]env-variable[=value]] ...
Context:	server config, virtual host, directory, .htaccess
Override:	FileInfo
Status:	Base
Module:	mod_setenvif

The `SetEnvIf` directive defines environment variables based on attributes of the request. The *attribute* specified in the first argument can be one of three things:

1. An HTTP request header field (see RFC2616² for more information about these); for example: Host, User-Agent, Referer, and Accept-Language. A regular expression may be used to specify a set of request headers.
2. One of the following aspects of the request:
 - Remote_Host - the hostname (if available) of the client making the request

Apache Module mod_setenvif

- `Remote_Addr` - the IP address of the client making the request
 - `Server_Addr` - the IP address of the server on which the request was received (only with versions later than 2.0.43)
 - `Remote_User` - the authenticated username (if available)
 - `Request_Method` - the name of the method being used (`GET`, `POST`, *et cetera*)
 - `Request_Protocol` - the name and version of the protocol with which the request was made (*e.g.*, "HTTP/0.9", "HTTP/1.1", *etc.*)
 - `Request_URI` - the resource requested on the HTTP request line -- generally the portion of the URL following the scheme and host portion without the query string
3. The name of an environment variable in the list of those associated with the request. This allows `SetEnvIf` directives to test against the result of prior matches. Only those environment variables defined by earlier `SetEnvIf[NoCase]` directives are available for testing in this manner. 'Earlier' means that they were defined at a broader scope (such as server-wide) or previously in the current directive's scope. Environment variables will be considered only if there was no match among request characteristics and a regular expression was not used for the *attribute*.

The second argument (*regex*) is a Perl compatible regular expression³. This is similar to a POSIX.2 `egrep`-style regular expression. If the *regex* matches against the *attribute*, then the remainder of the arguments are evaluated.

The rest of the arguments give the names of variables to set, and optionally values to which they should be set. These take the form of

1. *varname*, or
2. `!varname`, or
3. *varname=value*

In the first form, the value will be set to "1". The second will remove the given variable if already defined, and the third will set the variable to the literal value given by *value*.

Example:

```
SetEnvIf Request_URI "\.gif$" object_is_image=gif
SetEnvIf Request_URI "\.jpg$" object_is_image=jpg
SetEnvIf Request_URI "\.xbm$" object_is_image=xbm
:
SetEnvIf Referer www\.\mydomain\.com intra_site_referral
:
SetEnvIf object_is_image xbm XBIT_PROCESSING=1
:
SetEnvIf ^TS* ^[a-z].* HAVE_TS
```

The first three will set the environment variable `object_is_image` if the request was for an image file, and the fourth sets `intra_site_referral` if the referring page was somewhere on the `www.mydomain.com` Web site.

The last example will set environment variable `HAVE_TS` if the request contains any headers that begin with "TS" whose values begins with any character in the set `[a-z]`.

See also

- Environment Variables in Apache¹, for additional examples.

SetEnvIfNoCase Directive

Description:	Sets environment variables based on attributes of the request without respect to case
Syntax:	<code>SetEnvIfNoCase attribute regex [!]env-variable[=value]</code> <code>[[!]env-variable[=value]] ...</code>
Context:	server config, virtual host, directory, .htaccess
Override:	FileInfo
Status:	Base
Module:	mod_setenvif
Compatibility:	Apache 1.3 and above

The `SetEnvIfNoCase` is semantically identical to the `SetEnvIf` directive, and differs only in that the regular expression matching is performed in a case-insensitive manner. For example:

```
SetEnvIfNoCase Host Apache\.Org site=apache
```

This will cause the `site` environment variable to be set to "apache" if the HTTP request header field `Host:` was included and contained `Apache.Org`, `apache.org`, or any other combination.

URI References

- [1] <http://httpd.apache.org/docs-2.1/env.html>
- [2] <http://www.rfc-editor.org/rfc/rfc2616.txt>
- [3] <http://www.pcre.org/>