Terms Used to Describe Directives

This document describes the terms that are used to describe each Apache configuration directive[1].

Topics

See also

- Configuration files[2]

Description

A brief description of the purpose of the directive.

Syntax

This indicates the format of the directive as it would appear in a configuration file. This syntax is extremely directive-specific, and is described in detail in the directive's definition. Generally, the directive name is followed by a series of one or more space-separated arguments. If an argument contains a space, the argument must be enclosed in double quotes. Optional arguments are enclosed in square brackets. Where an argument can take on more than one possible value, the possible values are separated by vertical bars "|". Literal text is presented in the default font, while argument-types for which substitution is necessary are *emphasized*. Directives which can take a variable number of arguments will end in "..." indicating that the last argument is repeated.

Directives use a great number of different argument types. A few common ones are defined below.

URL
: A complete Uniform Resource Locator including a scheme, hostname, and optional pathname as in `http://www.example.com/path/to/file.html`

URL-path
: The part of a *url* which follows the scheme and hostname as in `/path/to/file.html`. The *url-path* represents a web-view of a resource, as opposed to a file-system view.

file-path
: The path to a file in the local file-system beginning with the root directory as in `/usr/local/apache/htdocs/path/to/file.html`. Unless otherwise specified, a *file-path* which does not begin with a slash will be treated as relative to the ServerRoot[3].

directory-path
: The path to a directory in the local file-system beginning with the root directory as in `/usr/local/apache/htdocs/path/to/`.

filename
: The name of a file with no accompanying path information as in `file.html`.

regex
: A regular expression, which is a way of describing a pattern to match in text. The directive

definition will specify what the *regex* is matching against.

extension
: In general, this is the part of the *filename* which follows the last dot. However, Apache recognizes multiple filename extensions, so if a *filename* contains more than one dot, each dot-separated part of the filename following the first dot is an *extension*. For example, the *filename* `file.html.en` contains two extensions: `.html` and `.en`. For Apache directives, you may specify *extension*s with or without the leading dot. In addition, *extension*s are not case sensitive.

MIME-type
: A method of describing the format of a file which consists of a major format type and a minor format type, separated by a slash as in `text/html`.

env-variable
: The name of an environment variable[4] defined in the Apache configuration process. Note this is not necessarily the same as an operating system environment variable. See the environment variable documentation[4] for more details.

Default

If the directive has a default value (*i.e.*, if you omit it from your configuration entirely, the Apache Web server will behave as though you set it to a particular value), it is described here. If there is no default value, this section should say "*None*". Note that the default listed here is not necessarily the same as the value the directive takes in the default httpd.conf distributed with the server.

Context

This indicates where in the server's configuration files the directive is legal. It's a comma-separated list of one or more of the following values:

server config
: This means that the directive may be used in the server configuration files (*e.g.*, `httpd.conf`), but **not** within any `<VirtualHost>` or `<Directory>` containers. It is not allowed in `.htaccess` files at all.

virtual host
: This context means that the directive may appear inside `<VirtualHost>` containers in the server configuration files.

directory
: A directive marked as being valid in this context may be used inside `<Directory>`, `<Location>`, and `<Files>` containers in the server configuration files, subject to the restrictions outlined in How Directory, Location and Files sections work[5].

.htaccess
: If a directive is valid in this context, it means that it can appear inside *per*-directory `.htaccess` files. It may not be processed, though depending upon the overrides currently active.

The directive is *only* allowed within the designated context; if you try to use it elsewhere, you'll get a configuration error that will either prevent the server from handling requests in that context correctly, or will keep the server from operating at all -- *i.e.*, the server won't even start.

The valid locations for the directive are actually the result of a Boolean OR of all of the listed contexts. In other words, a directive that is marked as being valid in "`server config, .htaccess`" can be used in the `httpd.conf` file and in `.htaccess` files, but not within any `<Directory>` or `<VirtualHost>` containers.

Override

This directive attribute indicates which configuration override must be active in order for the directive to be processed when it appears in a `.htaccess` file. If the directive's context doesn't permit it to appear in `.htaccess` files, then no context will be listed.

Overrides are activated by the `AllowOverride` directive, and apply to a particular scope (such as a directory) and all descendants, unless further modified by other `AllowOverride` directives at lower levels. The documentation for that directive also lists the possible override names available.

Status

This indicates how tightly bound into the Apache Web server the directive is; in other words, you may need to recompile the server with an enhanced set of modules in order to gain access to the directive and its functionality. Possible values for this attribute are:

Core
: If a directive is listed as having "Core" status, that means it is part of the innermost portions of the Apache Web server, and is always available.

MPM
: A directive labeled as having "MPM" status is provided by a Multi-Processing Module[6]. This type of directive will be available if and only if you are using one of the MPMs listed on the Module line of the directive definition.

Base
: A directive labeled as having "Base" status is supported by one of the standard Apache modules which is compiled into the server by default, and is therefore normally available unless you've taken steps to remove the module from your configuration.

Extension
: A directive with "Extension" status is provided by one of the modules included with the Apache server kit, but the module isn't normally compiled into the server. To enable the directive and its functionality, you will need to change the server build configuration files and re-compile Apache.

Experimental
: "Experimental" status indicates that the directive is available as part of the Apache kit, but you're on your own if you try to use it. The directive is being documented for completeness, and is not necessarily supported. The module which provides the directive may or may not be compiled in by default; check the top of the page which describes the directive and its module to see if it remarks on the availability.

Module

This quite simply lists the name of the source module which defines the directive.

Compatibility

If the directive wasn't part of the original Apache version 2 distribution, the version in which it was introduced should be listed here. In addition, if the directive is available only on certain platforms, it will be noted here.

URI References

[1] http://httpd.apache.org/docs-2.1/mod/directives.html
[2] http://httpd.apache.org/docs-2.1/configuring.html
[3] http://httpd.apache.org/docs-2.1/mod/core.html#serverroot
[4] http://httpd.apache.org/docs-2.1/env.html
[5] http://httpd.apache.org/docs-2.1/sections.html
[6] http://httpd.apache.org/docs-2.1/mpm.html