Configuration Files

This document describes the files used to configure Apache HTTP Server.

Main Configuration Files

Related Modules	Related Directives
mod_mime	<IfDefine> Include TypesConfig

Apache HTTP Server is configured by placing directives in plain text configuration files. The main configuration file is usually called `httpd.conf`. The location of this file is set at compile-time, but may be overridden with the `-f` command line flag. In addition, other configuration files may be added using the `Include` directive, and wildcards can be used to include many configuration files. Any directive may be placed in any of these configuration files. Changes to the main configuration files are only recognized by httpd when it is started or restarted.

The server also reads a file containing mime document types; the filename is set by the `TypesConfig` directive, and is `mime.types` by default.

Syntax of the Configuration Files

httpd configuration files contain one directive per line. The backslash "\" may be used as the last character on a line to indicate that the directive continues onto the next line. There must be no other characters or white space between the backslash and the end of the line.

Arguments to directives are separated by whitespace. If an argument contains spaces, you must enclose that argument in quotes.

Directives in the configuration files are case-insensitive, but arguments to directives are often case sensitive. Lines that begin with the hash character "#" are considered comments, and are ignored. Comments may **not** be included on the same line as a configuration directive. White space occurring before a directive is ignored, so you may indent directives for clarity. Blank lines are also ignored.

The values of variables defined with the `Define` or of shell environment variables can be used in configuration file lines using the syntax `${VAR}`.
If "VAR" is the name of a valid variable, the value of that variable is substituted into that spot in the configuration file line, and processing continues as if that text were found directly in the configuration file.
Variables defined with `Define` take precedence over shell environment variables.
If the "VAR" variable is not defined, the characters `${VAR}` are left unchanged, and a warning is logged. If instead a default value should be substituted, the conditional form `${VAR?=some default value}` can be used. Note that a **defined** empty variable will **not** be substituted with the default value, and that an empty default value like in `${VAR?=}` is a valid substitution (which produces an empty value if "VAR" is not defined, but no warning).
Variable names may not contain colon ":" characters, to avoid clashes with `RewriteMap`'s syntax.

Only shell environment variables defined before the server is started can be used in expansions. Environment variables defined in the configuration file itself, for example with `SetEnv`, take effect too late to be used for expansions in the configuration file.

The maximum length of a line in normal configuration files, after variable substitution and joining any continued lines, is approximately 16 MiB. In .htaccess files, the maximum length is 8190 characters.

You can check your configuration files for syntax errors without starting the server by using `apachectl configtest` or the `-t` command line option.

You can use mod_info's `-DDUMP_CONFIG` to dump the configuration with all included files and environment variables resolved and all comments and non-matching <IfDefine> and <IfModule> sections removed. However, the output does not reflect the merging or overriding that may happen for repeated directives.

Modules

Related Modules	Related Directives
mod_so	<IfModule>
	LoadModule

httpd is a modular server. This implies that only the most basic functionality is included in the core server. Extended features are available through modules which can be loaded into httpd. By default, a base set of modules is included in the server at compile-time. If the server is compiled to use dynamically loaded modules, then modules can be compiled separately and added at any time using the LoadModule directive. Otherwise, httpd must be recompiled to add or remove modules. Configuration directives may be included conditional on a presence of a particular module by enclosing them in an <IfModule> block. However, <IfModule> blocks are not required, and in some cases may mask the fact that you're missing an important module.

To see which modules are currently compiled into the server, you can use the `-l` command line option. You can also see what modules are loaded dynamically using the `-M` command line option.

Scope of Directives

Related Modules	Related Directives
	<Directory>
	<DirectoryMatch>
	<Files>
	<FilesMatch>
	<Location>
	<LocationMatch>
	<VirtualHost>

Directives placed in the main configuration files apply to the entire server. If you wish to change the configuration for only a part of the server, you can scope your directives by placing them in <Directory>, <DirectoryMatch>, <Files>, <FilesMatch>, <Location>, and <LocationMatch> sections. These sections limit the application of the directives which they enclose to particular filesystem locations or URLs. They can also be nested, allowing for very fine grained configuration.

httpd has the capability to serve many different websites simultaneously. This is called Virtual Hosting. Directives can also be scoped by placing them inside <VirtualHost> sections, so that they will only apply to requests for a particular website.

Although most directives can be placed in any of these sections, some directives do not make sense in some contexts. For example, directives controlling process creation can only be placed in the main server context. To find which directives

can be placed in which sections, check the Context of the directive. For further information, we provide details on How Directory, Location and Files sections work.

.htaccess Files

Related Modules	Related Directives
	AccessFileName
	AllowOverride

httpd allows for decentralized management of configuration via special files placed inside the web tree. The special files are usually called `.htaccess`, but any name can be specified in the `AccessFileName` directive. Directives placed in `.htaccess` files apply to the directory where you place the file, and all sub-directories. The `.htaccess` files follow the same syntax as the main configuration files. Since `.htaccess` files are read on every request, changes made in these files take immediate effect.

To find which directives can be placed in `.htaccess` files, check the Context of the directive. The server administrator further controls what directives may be placed in `.htaccess` files by configuring the `AllowOverride` directive in the main configuration files.

For more information on `.htaccess` files, see the .htaccess tutorial.