Apache HTTP Server Version 2.4

Getting Started

If you're completely new to the Apache HTTP Server, or even to running a website at all, you might not know where to start, or what questions to ask. This document walks you through the basics.

- Clients, Servers, and URLs
- Hostnames and DNS
- Configuration Files and Directives
- Web Site Content
- Log Files and Troubleshooting
- What's next?

See also

- Comments

Clients, Servers, and URLs

Addresses on the Web are expressed with URLs - Uniform Resource Locators - which specify a protocol (e.g. `http`), a servername (e.g. `www.apache.org`), a URL-path (e.g. `/docs/current/getting-started.html`), and possibly a query string (e.g. `?arg=value`) used to pass additional arguments to the server.

A client (e.g., a web browser) connects to a server (e.g., your Apache HTTP Server), with the specified protocol, and makes a **request** for a resource using the URL-path.

The URL-path may represent any number of things on the server. It may be a file (like `getting-started.html`) a handler (like server-status (# mod/mod_status.html)) or some kind of program file (like `index.php`). We'll discuss this more below in the Web Site Content (# #content) section.

The server will send a **response** consisting of a status code and, optionally, a response body. The status code indicates whether the request was successful, and, if not, what kind of error condition there was. This tells the client what it should do with the response. You can read about the possible response codes in HTTP Server wiki (# http://wiki. apache.org/httpd/CommonHTTPStatusCodes) .

Details of the transaction, and any error conditions, are written to log files. This is discussed in greater detail below in the Logs Files and Troubleshooting (# #logs) section.

Hostnames and DNS

In order to connect to a server, the client will first have to resolve the servername to an IP address - the location on the Internet where the server resides. Thus, in order for your web server to be reachable, it is necessary that the servername be in DNS.

If you don't know how to do this, you'll need to contact your network administrator, or Internet service provider, to perform this step for you.

More than one hostname may point to the same IP address, and more than one IP address can be attached to the same physical server. Thus, you can run more than one web site on the same physical server, using a feature called virtual hosts (# vhosts/) .

If you are testing a server that is not Internet-accessible, you can put host names in your hosts file in order to do local resolution. For example, you might want to put a record in your hosts file to map a request for `www.example.com` to your local system, for testing purposes. This entry would look like:

```
127.0.0.1 www.example.com
```

A hosts file will probably be located at `/etc/hosts` or `C:\Windows\system32\drivers\etc\hosts`.

You can read more about the hosts file at Wikipedia.org/wiki/Hosts_(file) (# http://en.wikipedia.org/wiki/Hosts_(file)) , and more about DNS at Wikipedia.org/wiki/Domain_Name_System (# http://en.wikipedia.org/wiki /Domain_Name_System) .

Configuration Files and Directives

The Apache HTTP Server is configured via simple text files. These files may be located any of a variety of places, depending on how exactly you installed the server. Common locations for these files may be found in the httpd wiki (# http://wiki.apache.org/httpd/DistrosDefaultLayout) . If you installed httpd from source, the default location of the configuration files is `/usr/local/apache2/conf`. The default configuration file is usually called `httpd.conf` . This, too, can vary in third-party distributions of the server.

The configuration is frequently broken into multiple smaller files, for ease of management. These files are loaded via the `Include` directive. The names or locations of these sub-files are not magical, and may vary greatly from one installation to another. Arrange and subdivide these files as makes the most sense to **you**. If the file arrangement you have by default doesn't make sense to you, feel free to rearrange it.

The server is configured by placing configuration directives (# mod/quickreference.html) in these configuration files. A directive is a keyword followed by one or more arguments that set its value.

The question of "*Where should I put that directive?*" is generally answered by considering where you want a directive to be effective. If it is a global setting, it should appear in the configuration file, outside of any `<Directory>`, `<Location>`, `<VirtualHost>`, or other section. If it is to apply only to a particular directory, then it should go inside a `<Directory>` section referring to that directory, and so on. See the Configuration Sections (# sections.html) document for further discussion of these sections.

In addition to the main configuration files, certain directives may go in `.htaccess` files located in the content directories. `.htaccess` files are primarily for people who do not have access to the main server configuration file (s). You can read more about `.htaccess` files in the `.htaccess` howto (# howto/htaccess.html) .

Web Site Content

Web site content can take many different forms, but may be broadly divided into static and dynamic content.

Static content is things like HTML files, image files, CSS files, and other files that reside in the filesystem. The `DocumentRoot` directive specifies where in your filesystem you should place these files. This directive is either set globally, or per virtual host. Look in your configuration file(s) to determine how this is set for your server.

Typically, a document called `index.html` will be served when a directory is requested without a file name being specified. For example, if `DocumentRoot` is set to `/var/www/html` and a request is made for `http://www.example.com/work/`, the file `/var/www/html/work/index.html` will be served to the client.

Dynamic content is anything that is generated at request time, and may change from one request to another. There are numerous ways that dynamic content may be generated. Various handlers (# handler.html) are available to generate content. CGI programs (# howto/cgi.html) may be written to generate content for your site.

Third-party modules like mod_php may be used to write code that does a variety of things. Many third-party applications, written using a variety of languages and tools, are available for download and installation on your Apache HTTP Server. Support of these third-party things is beyond the scope of this documentation, and you should find their documentation or other support forums to answer your questions about them.

Log Files and Troubleshooting

As an Apache HTTP Server administrator, your most valuable assets are the log files, and, in particular, the error log. Troubleshooting any problem without the error log is like driving with your eyes closed.

The location of the error log is defined by the `ErrorLog` directive, which may be set globally, or per virtual host. Entries in the error log tell you what went wrong, and when. They often also tell you how to fix it. Each error log message contains an error code, which you can search for online for even more detailed descriptions of how to address the problem. You can also configure your error log to contain a log ID which you can then correlate to an access log entry, so that you can determine what request caused the error condition.

You can read more about logging in the logs documentation (# logs.html) .

What's next?

Once you have the prerequisites under your belt, it's time to move on.

This document covers only the bare basics. We hope that this gets you started, but there are many other things that you might need to know.

- Download
- Install
- Configure
- Start
- Frequently Asked Questions

Comments

> **Notice:**
> This is not a Q&A section. Comments placed here should be pointed towards suggestions on improving the documentation or server, and may be removed by our moderators if they are either implemented or considered invalid/off-topic. Questions on how to manage the Apache HTTP Server should be directed at either our IRC channel, #httpd, on Libera.chat, or sent to our mailing lists.