# Apache HTTP Server Version 2.4

## Apache Module mod_vhost_alias

| | |
|---|---|
| **Description:** | Provides for dynamically configured mass virtual hosting |
| **Status:** | Extension |
| **Module Identifier:** | vhost_alias_module |
| **Source File:** | mod_vhost_alias.c |

### Summary

This module creates dynamically configured virtual hosts, by allowing the IP address and/or the `Host:` header of the HTTP request to be used as part of the pathname to determine what files to serve. This allows for easy use of a huge number of virtual hosts with similar configurations.

**Note**

If `mod_alias` or `mod_userdir` are used for translating URIs to filenames, they will override the directives of `mod_vhost_alias` described below. For example, the following configuration will map `/cgi-bin/script.pl` to `/usr/local/apache2/cgi-bin/script.pl` in all cases:

```
ScriptAlias "/cgi-bin/" "/usr/local/apache2/cgi-bin/"
VirtualScriptAlias "/never/found/%0/cgi-bin/"
```

## Directory Name Interpolation

All the directives in this module interpolate a string into a pathname. The interpolated string (henceforth called the "name") may be either the server name (see the `UseCanonicalName` directive for details on how this is determined) or the IP address of the virtual host on the server in dotted-quad format. The interpolation is controlled by specifiers inspired by `printf` which have a number of formats:

| | |
|---|---|
| `%%` | insert a % |
| `%p` | insert the port number of the virtual host |
| `%N.M` | insert (part of) the name |

`N` and `M` are used to specify substrings of the name. `N` selects from the dot-separated components of the name, and `M` selects characters within whatever `N` has selected. `M` is optional and defaults to zero if it isn't present; the dot must be present if and only if `M` is present. The interpretation is as follows:

| | |
|---|---|
| `0` | the whole name |
| `1` | the first part |
| `2` | the second part |
| `-1` | the last part |
| `-2` | the penultimate part |
| `2+` | the second and all subsequent parts |
| `-2+` | the penultimate and all preceding parts |
| `1+` and `-1+` | the same as `0` |

If `N` or `M` is greater than the number of parts available a single underscore is interpolated.

## Examples

For simple name-based virtual hosts you might use the following directives in your server configuration file:

```
UseCanonicalName    Off
VirtualDocumentRoot "/usr/local/apache/vhosts/%0"
```

A request for `http://www.example.com/directory/file.html` will be satisfied by the file `/usr/local/apache/vhosts/www.example.com/directory/file.html`.

For a very large number of virtual hosts it is a good idea to arrange the files to reduce the size of the `vhosts` directory. To do this you might use the following in your configuration file:

```
UseCanonicalName    Off
VirtualDocumentRoot "/usr/local/apache/vhosts/%3+/%2.1/%2.2/%2.3/%2"
```

A request for `http://www.domain.example.com/directory/file.html` will be satisfied by the file `/usr/local/apache/vhosts/example.com/d/o/m/domain/directory/file.html`.

A more even spread of files can be achieved by hashing from the end of the name, for example:

```
VirtualDocumentRoot "/usr/local/apache/vhosts/%3+/%2.-1/%2.-2/%2.-3/%2"
```

The example request would come from `/usr/local/apache/vhosts/example.com/n/i/a/domain/directory/file.html`.

Alternatively you might use:

```
VirtualDocumentRoot "/usr/local/apache/vhosts/%3+/%2.1/%2.2/%2.3/%2.4+"
```

The example request would come from `/usr/local/apache/vhosts/example.com/d/o/m/ain/directory/file.html`.

A very common request by users is the ability to point multiple domains to multiple document roots without having to worry about the length or number of parts of the hostname being requested. If the requested hostname is `sub.www.domain.example.com` instead of simply `www.domain.example.com`, then using %3+ will result in the document root being `/usr/local/apache/vhosts/domain.example.com/...` instead of the intended `example.com` directory. In such cases, it can be beneficial to use the

combination `%-2.0.%-1.0`, which will always yield the domain name and the tld, for example `example.com` regardless of the number of subdomains appended to the hostname. As such, one can make a configuration that will direct all first, second or third level subdomains to the same directory:

```
VirtualDocumentRoot "/usr/local/apache/vhosts/%-2.0.%-1.0"
```

In the example above, both `www.example.com` as well as `www.sub.example.com` or `example.com` will all point to `/usr/local/apache/vhosts/example.com`.

For IP-based virtual hosting you might use the following in your configuration file:

```
UseCanonicalName DNS
VirtualDocumentRootIP "/usr/local/apache/vhosts/%1/%2/%3/%4/docs"
VirtualScriptAliasIP  "/usr/local/apache/vhosts/%1/%2/%3/%4/cgi-bin"
```

A request for `http://www.domain.example.com/directory/file.html` would be satisfied by the file `/usr/local/apache/vhosts/10/20/30/40/docs/directory/file.html` if the IP address of `www.domain.example.com` were 10.20.30.40. A request for `http://www.domain.example.com/cgi-bin/script.pl` would be satisfied by executing the program `/usr/local/apache/vhosts/10/20/30/40/cgi-bin/script.pl`.

If you want to include the `.` character in a `VirtualDocumentRoot` directive, but it clashes with a `%` directive, you can work around the problem in the following way:

```
VirtualDocumentRoot "/usr/local/apache/vhosts/%2.0.%3.0"
```

A request for `http://www.domain.example.com/directory/file.html` will be satisfied by the file `/usr/local/apache/vhosts/domain.example/directory/file.html`.

The `LogFormat` directives %V and %A are useful in conjunction with this module.

## VirtualDocumentRoot Directive

| | |
|---|---|
| **Description:** | Dynamically configure the location of the document root for a given virtual host |
| **Syntax:** | `VirtualDocumentRoot` *`interpolated-directory`*`\|none` |
| **Default:** | `VirtualDocumentRoot none` |
| **Context:** | server config, virtual host |
| **Status:** | Extension |
| **Module:** | mod_vhost_alias |

The `VirtualDocumentRoot` directive allows you to determine where Apache HTTP Server will find your documents based on the value of the server name. The result of expanding *interpolated-directory* is used as the root of the document tree in a similar manner to the `DocumentRoot` directive's argument. If *interpolated-directory* is `none` then `VirtualDocumentRoot` is turned off. This directive cannot be used in the same context as `VirtualDocumentRootIP`.

> **Note**
>
> `VirtualDocumentRoot` will override any `DocumentRoot` directives you may have put in the same context or child contexts. Putting a `VirtualDocumentRoot` in the global server scope will effectively override `DocumentRoot` directives in any virtual hosts defined later on, unless you set `VirtualDocumentRoot` to `None` in each virtual host.

## VirtualDocumentRootIP Directive

| | |
|---|---|
| **Description:** | Dynamically configure the location of the document root for a given virtual host |
| **Syntax:** | `VirtualDocumentRootIP` *`interpolated-directory`*`\|none` |
| **Default:** | `VirtualDocumentRootIP none` |
| **Context:** | server config, virtual host |
| **Status:** | Extension |
| **Module:** | mod_vhost_alias |

The `VirtualDocumentRootIP` directive is like the `VirtualDocumentRoot` directive, except that it uses the IP address of the server end of the connection for directory interpolation instead of the server name.

## VirtualScriptAlias Directive

| | |
|---|---|
| **Description:** | Dynamically configure the location of the CGI directory for a given virtual host |
| **Syntax:** | `VirtualScriptAlias` *`interpolated-directory`*`\|none` |
| **Default:** | `VirtualScriptAlias none` |
| **Context:** | server config, virtual host |
| **Status:** | Extension |
| **Module:** | mod_vhost_alias |

The `VirtualScriptAlias` directive allows you to determine where Apache httpd will find CGI scripts in a similar manner to `VirtualDocumentRoot` does for other documents. It matches requests for URIs starting `/cgi-bin/`, much like `ScriptAlias /cgi-bin/` would.

## VirtualScriptAliasIP Directive

| | |
|---|---|
| **Description:** | Dynamically configure the location of the CGI directory for a given virtual host |
| **Syntax:** | `VirtualScriptAliasIP` *`interpolated-directory`*`\|none` |
| **Default:** | `VirtualScriptAliasIP none` |
| **Context:** | server config, virtual host |
| **Status:** | Extension |
| **Module:** | mod_vhost_alias |

The `VirtualScriptAliasIP` directive is like the `VirtualScriptAlias` directive, except that it uses the IP address of the server end of the connection for directory interpolation instead of the server name.

## 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.