

## Apache Module `mod_vhost_alias`

<b>Description:</b>	Provides for dynamically configured mass virtual hosting
<b>Status:</b>	Extension
<b>Module Identifier:</b>	<code>vhost_alias_module</code>
<b>Source File:</b>	<code>mod_vhost_alias.c</code>

### 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/
```

### Topics

Directory Name Interpolation.....	1
Examples .....	2
URI References .....	4

### Directives

<code>VirtualDocumentRoot</code> .....	3
<code>VirtualDocumentRootIP</code> .....	3
<code>VirtualScriptAlias</code> .....	3
<code>VirtualScriptAliasIP</code> .....	4

### See also

- [UseCanonicalName](#)
- Dynamically configured mass virtual hosting<sup>1</sup>

## 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:

<code>%%</code>	insert a %
<code>%p</code>	insert the port number of the virtual host
<code>%N.M</code>	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

---

 Apache Module mod\_vhost\_alias
 

---

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`.

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

---

 Apache Module mod\_vhost\_alias
 

---

```
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.com/directory/file.html`.

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

## VirtualDocumentRoot Directive

---

<b>Description:</b>	Dynamically configure the location of the document root for a given virtual host
<b>Syntax:</b>	<code>VirtualDocumentRoot <i>interpolated-directory</i> none</code>
<b>Default:</b>	<code>VirtualDocumentRoot none</code>
<b>Context:</b>	server config, virtual host
<b>Status:</b>	Extension
<b>Module:</b>	mod_vhost_alias

The `VirtualDocumentRoot` directive allows you to determine where Apache 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`.

## VirtualDocumentRootIP Directive

---

<b>Description:</b>	Dynamically configure the location of the document root for a given virtual host
<b>Syntax:</b>	<code>VirtualDocumentRootIP <i>interpolated-directory</i> none</code>
<b>Default:</b>	<code>VirtualDocumentRootIP none</code>
<b>Context:</b>	server config, virtual host
<b>Status:</b>	Extension
<b>Module:</b>	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

---

<b>Description:</b>	Dynamically configure the location of the CGI directory for a given virtual host
<b>Syntax:</b>	VirtualScriptAlias <i>interpolated-directory</i>  none
<b>Default:</b>	VirtualScriptAlias none
<b>Context:</b>	server config, virtual host
<b>Status:</b>	Extension
<b>Module:</b>	mod_vhost_alias

The `VirtualScriptAlias` directive allows you to determine where Apache 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

---

<b>Description:</b>	Dynamically configure the location of the cgi directory for a given virtual host
<b>Syntax:</b>	VirtualScriptAliasIP <i>interpolated-directory</i>  none
<b>Default:</b>	VirtualScriptAliasIP none
<b>Context:</b>	server config, virtual host
<b>Status:</b>	Extension
<b>Module:</b>	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.

## URI References

---

- [1] <http://httpd.apache.org/docs-2.1/vhosts/mass.html>