CanonicalHostNames

Canonical HostNames

Many sites map a variety of hostnames to the same content. For example, example.com, www.example.com and www.example.net may all refer to the same site. It is best to make sure that, regardless of the name clients use to access the site, they will be redirected to a single, canonical hostname, preferably without "www.". This makes the site easier to maintain and assures that there will be only one version of the site in proxy caches and search engines.

Using Virtual Hosts

Use the Redirect directive inside a `VirtualHost` block to force browsers to redo a request with a new host name.

Remember that the first listed `VirtualHost` will serve all requests that match no `ServerName` or `ServerAlias`. By using a Redirect in this host, you can assure that any non-matched names will also be transferred to the Canonical host.

```
# Required for 2.2.x and below:
#NameVirtualHost *:80

# www.example.net and www.example.com VirtualHost
<VirtualHost *:80>
  ServerName www.example.net
  ServerAlias www.example.com
  Redirect permanent / http://example.com/
</VirtualHost>

# Canonical VirtualHost
<VirtualHost *:80>
  ServerName example.com
  DocumentRoot /usr/local/apache/htdocs
</VirtualHost>
```

Using mod_rewrite

```
# For sites running on a port other than 80
RewriteCond %{HTTP_HOST}   !^example\.com [NC]
RewriteCond %{HTTP_HOST}   !=""
RewriteCond %{SERVER_PORT} !=80
RewriteRule ^/?(.*)         http://example.com:%{SERVER_PORT}/$1 [L,R=301]

# And for a site running on port 80
RewriteCond %{HTTP_HOST}   !^example\.com [NC]
RewriteCond %{HTTP_HOST}   !=""
RewriteRule ^/?(.*)         http://example.com/$1 [L,R=301]
```