

DebianDeb0rkification

Debian Deb0rkification

This document is meant to be used by new debian and ubuntu users. It describes the various differences between upstream sources and the modifications made by the debian maintainers.

Please note that this is a work in progress, as Debian's configs are a moving target. Any comments are most welcome.

A more comprehensive list of changes can be found in `/usr/share/doc/apache2.2-common/README.Debian.gz`

Like many articles here, the main purpose of this one is to ease the difficulties arising from supporting new users in `#httpd`. One of the main causes of headaches there is people who are new to both Apache HTTPd and their Debian Linux system.

This guide describes how to make efficient use of the Debian configuration files, while preserving the various automation they provide. Please note that we will be working with the latest stable release.

apache2.conf

Debian calls its main configuration file *apache2.conf*, which is the first source of confusion, because there is also an *httpd.conf* file in the configuration directory:

```
jmcg@panic:/etc/apache2$ ls -ltr
total 48
-rw-r--r-- 1 root root 59 2008-01-17 22:26 ports.conf
-rw-r--r-- 1 root root 378 2008-01-17 22:26 envvars
-rw-r--r-- 1 root root 0 2008-01-23 08:12 httpd.conf
drwxr-xr-x 2 root root 4096 2008-03-31 13:37 sites-enabled
-rw-r--r-- 1 root root 10826 2008-05-14 01:35 apache2.conf
drwxr-xr-x 2 root root 4096 2008-06-11 14:35 sites-available
drwxr-xr-x 2 root root 4096 2008-06-11 14:35 conf.d
drwxr-xr-x 2 root root 4096 2008-06-11 14:35 mods-enabled
drwxr-xr-x 2 root root 12288 2008-06-11 14:35 mods-available
```

But a quick *apache2 -V* reveals, among other useful things, what the configuration file is:

```
jmcg@panic:/etc/apache2$ sudo apache2 -V
Server version: Apache/2.2.8 (Debian)
Server built: May 13 2008 23:39:43
Server's Module Magic Number: 20051115:11
Server loaded: APR 1.2.12, APR-Util 1.2.12
Compiled using: APR 1.2.12, APR-Util 1.2.12
Architecture: 64-bit
Server MPM: Prefork
  threaded: no
  forked: yes (variable process count)
Server compiled with....
-D APACHE_MPM_DIR="server/mpm/prefork"
-D APR_HAS_SENDFILE
-D APR_HAS_MMAP
-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
-D APR_USE_SYSVSEM_SERIALIZE
-D APR_USE_PTHREAD_SERIALIZE
-D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
-D APR_HAS_OTHER_CHILD
-D AP_HAVE_RELIABLE_PIPED_LOGS
-D DYNAMIC_MODULE_LIMIT=128
-D HTTPD_ROOT=""
-D SUEXEC_BIN="/usr/lib/apache2/suexec"
-D DEFAULT_PIDLOG="/var/run/apache2.pid"
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
-D DEFAULT_LOCKFILE="/var/run/apache2/accept.lock"
-D DEFAULT_ERRORLOG="logs/error_log"
-D AP_TYPES_CONFIG_FILE="/etc/apache2/mime.types"
-D SERVER_CONFIG_FILE="/etc/apache2/apache2.conf"
```

The main problem with the *apache2.conf* file is not its name, but the lack of sane defaults, which we will now rectify:

```

## snip
# Change the default of 15 seconds to a more conservative value:
KeepAliveTimeout 2

## snip

# Leave this as is, as it makes sense (see below)
# maybe change the name of the file..
#
# Define an access log for VirtualHosts that don't define their own logfiles
CustomLog /var/log/apache2/other_vhosts_access.log vhost_combined

# For reasons of paranoia, set this to Prod, instead of Debian's default 'Full'
ServerTokens Prod

# Same here; no one needs to know our version number:
ServerSignature Email

# This is what will be returned by the above:
ServerAdmin webmaster@localhost

# Only disable this if you really want to ;)
TraceEnable Off

# This is for usability:
AcceptPathinfo On

# Debian puts this in their overly verbose Default VHost, but that's unnecessary. A single instance before all
the virtual hosts is sufficient
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

# Assuming /var/www/ is the root of all of your vhosts, set sane defaults for it:
<Directory /var/www/*/htdocs>
    Options +MultiViews
# The user may add +Indexes if he wishes to display a formatted list of the directory contents in the absence
of index files as specified with the ''DirectoryIndex'' directive.
    Allow from All
    AllowOverride None
</Directory>

```

And that's it. These additions/changes in *apache2.conf* set sane defaults.

sites-enabled

There's one marvelous attribute of Debian's configuration: the introduction of a default virtual host, and scripts to automate the addition/removal of *sites*.

The default *sites* file, called */etc/apache2/sites-available/default/000-default*, contains this:

```

NameVirtualHost *:80
<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

    ErrorLog /var/log/apache2/error.log

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn

    CustomLog /var/log/apache2/access.log combined
    ServerSignature On

    Alias /doc/ "/usr/share/doc/"
    <Directory "/usr/share/doc/">
        Options Indexes MultiViews FollowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 127.0.0.0/255.0.0.0 ::1/128
    </Directory>
</VirtualHost>

```

This configuration is overly verbose, complex, and unnecessary. Instead of pointing out every configuration error in it, I will rewrite it from scratch: In *apache2.conf* change:

```

# Include all the user configurations:
# BUT not before declaring that we will be using Name-based vhosts!
NameVirtualHost *:80
Include /etc/apache2/httpd.conf

```

Also Note that in newer versions of Ubuntu/Debian *NameVirtualHost *:80* has been moved to *ports.conf*.

```

<VirtualHost *:80>
    ServerName some.domain.tld
    DocumentRoot /var/www/some.domain.tld/htdocs

    ErrorLog /var/log/apache/some.domain.tld/error_log
</VirtualHost>

```

We removed the [CustomLog](#) directive as the one defined in *apache2.conf* will propagate to every virtual host - effectively reducing the number of open file handles.

We got rid of all the superfluous *<Directory /var/www>* and *<Directory />* blocks - especially with its awkward *Options* directives. Again, with the sane settings in the *apache2.conf* file.

We defused the danger of ambiguity, by changing and moving the *NameVirtualHost *:80* directive above the *Include* line for the virtual hosts, thus enabling new administrators to simply copy, paste, and edit this file.

mods-enabled

a2enmod and *a2dismod* can be used to link configuration sinplets from *mods-available* to *mods-enabled* which then get's included into *apache2.conf*. This is supposed to ease administration. Yet special caution has to be taken with certain modules when enabled in Debian. A wiki page has already been devoted to [PHP](#).

Another module worth mentioning is *mod_proxy*. It's default configuration is that of a Reverse-Proxy, and this is emphasized in Debian's default config:

```
<IfModule mod_proxy.c>
    #turning ProxyRequests on and allowing proxying from all may allow
    #spammers to use your proxy to send email.

    ProxyRequests Off

    <Proxy *>
        AddDefaultCharset off
        Order deny,allow
        Deny from all
        #Allow from .example.com
    </Proxy>

    # Enable/disable the handling of HTTP/1.1 "Via:" headers.
    # ("Full" adds the server version; "Block" removes all outgoing Via: headers)
    # Set to one of: Off | On | Full | Block

    ProxyVia On
</IfModule>
```

Now the problem here is the directive *Deny from all*, which effectively denies access to the proxy, leading to 403 Errors [ClientDeniedByServerConfiguration](#) errors in the [ErrorLog](#).