

# ApacheVirtualHostMysql

## Introduction

Here you'll find an example of how to do name-based virtual hosting using mod\_perl and mysql.

This works for my needs but most likely you will have to change parts to get what you want. Everything covered here is fairly advanced and I'll assume you know your way around Apache, MySQL and Perl.

I tried to make it as portable as possible. It should work with a minimum changes on any Linux system, and with more on Windows as well.

**Note: If you experience problems you can try to contact me on #httpd (ask for sjorge)**

## Requirements

1. perl 2. httpd 2.2
  - a. mod\_perl 2. mod\_dav / mod\_dav\_fs 3. mod\_authn\_dbd 4. mod\_ftp (can be left out – changes to the files are needed) 5. libphp5 (can be left out – changes to the files are needed)
3. MySQL server (another database server will work if there are DBI and DBD drivers available – changes to the files are needed)

## General Setup

Configuration Layout

path	description
/srv/httpd	<a href="#">ServerRoot</a>
/srv/httpd/conf/httpd.conf	Main configuration file
/srv/httpd/conf/hosts.conf	Host configuration file
/srv/httpd/conf/template/http.tpl	Virtual host template
/srv/httpd/conf/template/webdav.tpl	WebDAV virtual host template
/srv/httpd/conf/template/ftp.tpl	FTP virtual host template
/srv/hosts	hosts base path

Virtual Host Layout (relative to /srv/hosts)

path	description
<i>\$name</i>	Virtual host base dir
<i>\$name/_sys</i>	Holds varies other files
<i>\$name/_sys/logs</i>	Logs for host
<i>\$name/_sys/tmp</i>	temp directory for php
<i>\$name/_sys/sessions</i>	sessions directory for php
<i>\$name/cgi-bin</i>	cgi-bin for host (only works when CGI is set in database)
<i>\$name/httpdocs</i>	document root

## Setting up the database

First we need a database to hold our host configuration.

Use the MySQL client, or phpMyAdmin, or some other means, to create the database. You can get the database template [here](#)

You also need to create a user with access to that database. I chose to call mine **apache**

- more information on the database will follow later

## Configuring httpd

**Note: Replace /srv/httpd with your own ServerRoot!!**

Download all .conf and .tmpl files provided in the **General Setup** section and save them in the correct locations.

Edit httpd.conf and change:

- [ServerRoot](#)
- [ServerName](#)
- also update all /srv/httpd occurrences to your [ServerRoot](#)

Edit hosts.conf and change:

- db server
- db database
- db user
- db pass
- host http\_tmpl path (full path!)
- host dav\_tmpl path (full path!)
- host ftp\_tmpl path (full path!)
- host path (full path!)

Edit http.tmpl and webdav.tmpl and change:

- all /srv/httpd occurrences to your [ServerRoot](#)

You should be good to go. Feel free to make more changes to the templates and configuration files

## Adding a host

Use the MySQL client or phpMyAdmin, or another means, to access the database.

### adding host entry

```
INSERT INTO hosts VALUES(null, 'localhost', 1, 0, 0, 0, 0, 0);
```

value	description
null	auto-increment host ID
'localhost'	host name
1	enable (0 to disable the host)
0	disable WebDAV (1 to enable)
0	disable FTP (1 to enable)
0	disable CGI (1 to enable)
0	disable Server Side Includes (1 to enable)
0	disable PHP (1 to enable)

(Re)start httpd and the new vhosts will be available (you need to provide the -DDAV flag to enable WebDAV on the server)

**Note: The directories for this vhost will be created automatically when apache is (re)started! (There may be a spike in memory usage at startup.)**

**Note: Removal isn't automatic; if you remove a host + aliases + users + configuration the files will remain)**

**Note: I recommend adding localhost as your first vhost. All request with and unknown host header will go to this host.**

### adding users

Only required when you enable WebDAV for a host.

```
INSERT INTO users VALUES(null, 1, 'user', 'apache-encrypted-password', 'dav');
```

value	description
null	auto increment user id
1	host id user belongs too (check hosts table to get it)
'user'	user name

'apache-encrypted-password'	password for user. check httpd doc for encryption ( <a href="#">here</a> )
'dav'	group(s) for user. dav = enables WebDAV access

## adding aliases (optional)

You can create aliases for hosts as well if you wish (multiple aliases possible per host; each alias must be unique).

```
INSERT INTO aliases VALUES(1, 'cygnus');
```

value	description
1	host ID alias belongs too (check hosts table to get it)
'cygnus'	alias for host

## adding additional configuration options (optional)

Setting up a host as reverse proxy? not a problem (note reverse proxy config not shown).

```
INSERT INTO configuration VALUES(1, '#proxy config');
```

*value_	• description_
1	host id configuration belongs too (check hosts table to get it)
'#proxy config'	configuration for host

## Alternative Methods of managing vhosts

There is a google code project that has a java based manager for this configuration. Use at own risk. This isn't professional code and isn't tied to the ASF.

<http://code.google.com/p/vhm-mysql/>

## accessing WebDAV

Point your WebDAV client to <http://www.example.com:81/httpdocs/> and long in with the user/pass combination. It works in IE, [DreamWeaver](#), cadaver, and Mac OS X's WebDAV client.

## accessing FTP

This is a bit more tricky.

```
host: www.example.com
user: user@www.example.com
pass: pass
```

The vhost is selected based on the username. A username without a suffix will go to the first vhost with FTP enabled.

## Portability

This configuration works on both Windows and Linux (presumably Macs as well).

To get it to work on Windows you need to edit http.tpl and update the path separate from : to ; for the php\_admin\_values.

Other database systems can be used by modifying the configuration files. PostgreSQL should work fine, for example.