



# **Windows Web Server Configuration**

Installation Guide for Apache 2.0.x Web Server

WAMP

Version: 1.6  
Author: Stamati Crook  
redware research ltd  
<http://www.redware.com>  
Date: 29 January 2005  
Document: apache0106.doc

# Contents

Windows Web Server Configuration.....	1
Contents .....	2
Copyright © 2004, 2005. ....	3
Introduction .....	3
Quick Installation .....	3
Apache Installation .....	4
Default Configuration File.....	5
Stop and Start Apache.....	6
Create a Minimal Configuration .....	6
Load Optional Modules.....	7
Configure MIME types .....	7
Set up log files .....	8
Performance Directives .....	9
Directories.....	10
Default Settings .....	10
Directory Settings .....	11
Create user directories .....	12
Create a private directory .....	12
Applications .....	13
Activate Server Side Includes .....	13
Install MySQL.....	13
Create a new MySQL database and user .....	14
Backup a mysql database .....	14
Install PHP.....	14
Install Web Applications .....	15
Security.....	16
Linux Notes.....	16
Conclusion .....	17
References .....	18
httpd.conf .....	18

## Copyright © 2004, 2005.

The document is brought to you by **redware research ltd** and is copyright and published commercially. This means that you cannot distribute the document freely and should instead refer colleagues to our web site or to <http://www.amazon.com> where they may download their own copy for a small fee. Although every precaution has been taken in the preparation of this document, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein. For information, please contact:

**redware research ltd, 104 Tamworth Road, Hove BN3 5FH, England.**  
<http://www.redware.com>

This document is updated regularly and we would be pleased to send you an update if you register on our website. You will need this password: `apache05`. We also have other documents covering database development with Visual FoxPro and SQL Server available from our website and [amazon.com](http://www.amazon.com).

## Introduction

Apache is a world class web server that can run on most personal computers on several operating systems including Microsoft Windows, Linux, and Macintosh OSX. Linux is the most robust and secure operating system for Apache and has become the industry standard web server for very large and very busy commercial web sites as well as being used by scores of web hosting providers.

Apache can be installed on a standard personal computer and provides a great web server on a standard Windows machine. This document and the associated configuration file allows you to instantly create a simple, useful, and secure web server installation on your Windows PC with the minimum of fuss and bother.

The default Apache installation provides an excellent default configuration which can be extended immediately by experienced administrators. However the configuration is complex and difficult to understand for novice administrators. This installation guide helps you to build a configuration file that contains only the features you need for your installation and places you in a good position to manage your web server installation as you add additional features over time.

The **redware** configuration of Apache 2.0.x has the following features:

- The configuration and log files are maintained in the default sub-folders of the Apache installation.
- Web content and security configuration is placed in a convenient folder for regular backup.
- A separate user folder allows easy management of individual user or departmental web content.
- A private folder allows username and password access to confidential web pages.
- PHP and MySQL installation allows LAMP (or WAMP) applications to be installed easily.

Default security settings are specified to provide a reasonable level of security. Compilation of Apache from source code is not covered and consequently you will have problems setting up SSL security on your Windows web server as Apache exclude the relevant program from the Windows binary distribution. PHP and MySQL installation is briefly covered so that you can easily install web applications that conform to the LAMP standard. However be warned that the PHP people warn that Apache 2.0 is not suitable for a production server and you should consider Apache 1.3.x on Linux.

These notes are concise and provide the basic information you need to know to understand your server configuration. Please consult the on-line Apache documentation at <http://httpd.apache.org/docs-2.0/> for further information.

## Quick Installation

A quick installation of the **redware** Apache web server configuration involves the following stages:

- Download and install the Windows binary version of Apache 2.0.x from the [www.apache.org](http://www.apache.org) web site. You will be prompted for a domain name, a server name, and an email address for the administrator. If your web server does not have a valid domain name then leave this blank or use the IP address of the machine for the server name.

- Test the installation by browsing **http://localhost**.
- Download the `web.zip` file from our download pages at **http://www.redware.com/download.html** and unzip into the `c:\web` folder (or similar) on your PC.
- Rename the original `httpd.conf` configuration file from the `conf` sub-folder of the apache installation (normally held in `c:\program files\apache group\apache2\conf`) and overwrite with the redware configuration file found in `c:\web\httpd.conf` (and also listed at the end of this document).
- Restart the apache server from the Windows Control Panel or the Apache Monitor present on your toolbar or with the following commands from the DOS command prompt.
 

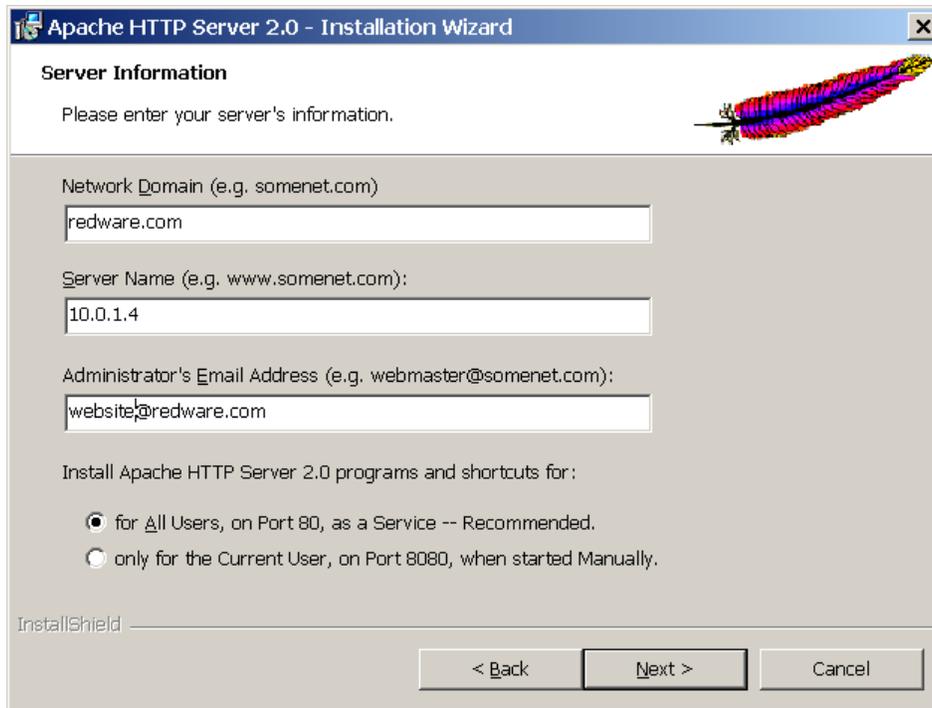
```
c:
cd "\program files\apache group\apache2\"
bin\apache -k restart
```
- Test the server by browsing **http://localhost**. Look in the `logs\error.log` file for error messages if there are any problems.
- Add your own html files into the `c:\web\htdocs` folder perhaps with a default `index.html` page.
- Now read the rest of this document so you can understand your installation.

## Apache Installation

The easiest installation for Windows employs the Windows 32 binary installation available from the download page at **http://httpd.apache.org/download.cgi**. Make sure you download the latest recommended version of Apache 2.0 and not Apache 1.3.

The installation is straightforward and requires the following information:

- Domain name of the computer. If you do not have a domain then leave this blank.
- Use the IP address of your machine if you do not have a valid DNS name for the server. This can be found by typing `ipconfig` at the DOS command prompt. Note that your IP address can change each time you reboot your machine if your network connection is configured to use a DHCP server. You should ask your system administrator to assign you a static IP address so your machine is visible at the same address after a reboot.
- Administrator's email may appear on some error pages generated by the server.



### *Windows Installation*

Apache should run automatically after installation and you can check for a successful installation by browsing the default page on your local machine using the following URL <http://localhost>. Test from other machines by using the IP address of the host machine, for example, <http://10.0.1.4>.

Apache is installed in the folder you specified or at `c:\program files\apache group\apache2`. The subfolders contain the configuration and content for the server and include the following:

- `bin` contains the Apache program and some useful utilities.
- `bin\modules` contains additional pre-compiled modules that can be included by altering your configuration.
- `conf` contains the configuration files.
- Logs are stored in the `logs` subfolder.
- `htdocs` contains the web pages served by the server.
- Additional folders contain the documentation and other features.

### **Default Configuration File**

A sophisticated and extensive configuration file is created automatically in the `conf` subfolder of your installation offering many configuration features including many of the features described in this document and:

- Fancy indexing with icons displayed for file types.
- Multiple languages to display language specific pages according to the browser setup.
- Multiple character sets and encodings.

Our approach is to create a blank configuration file and adopt an incremental approach to implementing additional features. This results in a configuration file that you can understand and implements required features only. Additional features from the default configuration file can be added at a later date by cutting and pasting between the files. This approach can also be used to test new features in a simple context before copying the relevant configuration into the main configuration file.

The administrator of a high performance web server might be advised to use and modify the default configuration file or at least read through and understand the implemented features. Your final configuration file is similar to the installation configuration once you have worked through and implemented all the features discussed in this guide.

## Stop and Start Apache

Problems often occur during the start up process when changing and testing the configuration files. These are logged to an error file or to the Windows event log and are not immediately visible if you are using the graphical Apache Monitor interface. A cleaner approach is to use the command line interface to stop and start Apache so that error messages are instantly visible.

Start the operating system console in Windows (perhaps by selecting `RUN...` from the start menu and typing `cmd`) and use the command line to stop the web server after changing to the Apache installation folder:

```
c:
cd \program files\apache group\apache2\
bin\apache -k stop
```

And start as follows:

```
bin\apache -k start
```

The `restart` option is useful to restart Apache gracefully if there are existing users accessing the web server (but can be confusing when testing your configurations):

```
bin\apache -k restart
```

You can also have a selection of configuration files and get Apache to start with a specific configuration file using the following option (remember to use `/` instead of `\` in the path description):

```
bin\apache -k start -f c:/web/httpd-minimal.conf
```

A listing of the options available is obtainable:

```
bin\apache -help
```

In Windows, the Apache service is installed with the name **apache2** which you can also control from the Services window available from the Control Panel (Administration Tools) or from the command line with following commands:

```
net stop apache2
net start apache2
```

You can install the service if you have installed Apache without using the installation program or there is no service installed with the following command:

```
bin\apache -k install
```

## Create a Minimal Configuration

A minimal configuration file is shown below and can replace the default `httpd.conf` by starting Apache from the command line as follows:

```
bin\apache -k restart -f c:/web/httpd-minimal.conf
```

The configuration file will need to be changed for your installation:

- `ServerName` needs your server IP address or a valid DNS entry. It is better to leave this blank than enter an incorrect value.
- `ServerRoot` is your Apache installation folder and allows Apache to find the default configuration and log files.
- `DocumentRoot` is the folder used to contain the web server content. The existing `htdocs` folder in the Apache installation folder is now ignored.
- `ServerAdmin` should refer to a valid email address.

*Note: Paths for Windows need to be enclosed in double quotes and use a UNIX-style forward slash instead of a backslash. The Windows version of Apache recognises the drive letter prefix.*

Test the server with the following URL and note that the `index.html` file must be fully specified as there is no default page for this configuration:

```
http://localhost/index.html
```

Don't worry if the page is displayed as text. The section on configuring MIME types will resolve this problem. The minimal configuration is listed below and contained in the `web.zip` download:

```
# bof httpd-minimal.conf
# redware apache minimal configuration
# Version 1.5 30 December 2004
# Author Stamati Crook redware research ltd
# www.redware.com

##### MINIMAL
# Listen out for browsers on port 80.
Listen 80

# ServerName and Port used to create redirection URLs.
# Should be a valid DNS name or the IP address if there is no DNS.
# Let Apache work out the ServerName.
# ServerName

# Location for conf, error, and log folders (no trailing slash).
ServerRoot "c:/Program Files/Apache Group/Apache2"

# Location of html pages (no trailing slash)
DocumentRoot "c:/web/htdocs"

# Administrator's email
ServerAdmin website@redware.com

# eof httpd-minimal.conf
```

## Load Optional Modules

Many of the configuration options described in the remainder of the document require additional modules to be loaded as Apache begins execution. The binary version of Apache has many of these modules compiled into the `modules` folder and the `LoadModule` command is all that is required to enable these options.

A common error that prevents Apache from starting is the specification of a configuration option without the appropriate module being loaded. The error message looks like this:

```
Invalid command 'DirectoryIndex', perhaps mis-spelled or defined by a module not
included in the server configuration
```

The appropriate module needs to be added to the configuration file to remove this error as in the following example:

```
LoadModule dir_module modules/mod_dir.so
```

The `LoadModule` commands should be placed near the beginning of the configuration file to make sure that they are loaded before they are required by a configuration directive.

*Note: The `mod_ssl` module that provides the SSL transport layer to allow your web server to handle https: requests is not included with the Windows binary. This makes installing SSL on Windows a difficult process which is not covered in this document.*

## Configure MIME types

Web browsers associate particular files with particular applications by communicating the MIME type of the file to the browser. The following options allow Apache to determine the MIME type of many different file extensions and communicate this to the browser.

```
##### MIME TYPES
```

```

<IfModule !mod_mime.c>
  # Allows TypesConfig to work
  LoadModule mime_module modules/mod_mime.so
</IfModule>

DefaultType text/plain

# Map files to mime types
# Required for Mozilla Firefox to display html files as html and not text.
TypesConfig conf/mime.types

# Add the mime_magic module if more mime types are needed.
#<IfModule mod_mime_magic.c>
#   MIMEMagicFile conf/magic
#</IfModule>

```

The `DefaultTypes` and `TypesConfig` allow Apache to provide information on the file types for various file extensions so that the browser can format the content appropriately. Note that the `mod_mime.so` module must be loaded by the `LoadModule` command for the `TypesConfig` command to function.

*Note: Many MIME types are specified automatically but you can add your own associations with the `AddType` directive or add the `Mime_Magic` module..*

## Set up log files

The following configuration commands allow the Apache server to log activity and errors to the appropriate files in the `logs` folder of the apache installation:

- `ErrorLog` allows Apache to log errors into the specified text file (perhaps in addition to the Windows event log).
- The `PidFile` setting logs an operating system process identifier for the Apache program (not useful in Windows).
- The `CustomLog` and `LogFormat` commands specify the log file and the format for the pages requested of the server. This file can be processed by a number of log file analysers to give a report of activity on the server.

```

##### LOGS

<IfModule !mod_log_config.c>
  # Needed for LogFormat
  LoadModule log_config_module modules/mod_log_config.so
</IfModule>

# Errorlog (relative to ServerRoot).
ErrorLog logs/error.log

#PidFile not really useful for Windows - used to kill process in UNIX.
PidFile logs/http.pid

# Common Log format (not used).
#Log File in the Common Log file format.
#LogFormat "%h %l %u %t \"%r\" %>s %b" common
#CustomLog logs/access.log common

```

```
# Combined log format.
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
#CustomLog logs/access.log combined

# Optionally rotate logs to a new file every 24 hours.
CustomLog "|bin/rotatelogs.exe logs/log%Y%m%d%H.log 86400" combined
```

The `common` format is the default format acceptable to all log file analysers but the `combined` format has additional information for the referrer and browser type is specified in this configuration. The referrer is important to determine sites or search pages that link to your site and can also be analysed by many log analysers.

*Note: The second parameter in the log files can be used to determine an identify flag for the browser but is unreliable and causes a performance overhead and is therefore switched off by the `IdentityCheck` setting.*

## Performance Directives

The following directives should be set to ensure the correct performance tuning from your Windows web server. Please consult the documentation before tuning any of these parameters.

```
##### PERFORMANCE

# http network level performance settings
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

# WinNT MPM - specific windows recommended settings.
# ThreadsPerChild: constant number of worker threads in the server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_winnt.c>
    ThreadsPerChild 50
    MaxRequestsPerChild 0
</IfModule>

# Off stops apache substituting ServerName and Port for redirection.
UseCanonicalName Off

# Unreliable way to get user name from the browser so not used.
IdentityCheck off

# Make sure DNS lookups are disabled (this is the default).
HostnameLookups Off

# Performance notes:
# Set MaxClients if your server has too many users.
# Add the browsermatch directives from the default configuration.
# Different settings required for Linux.
```

# Directories

## Default Settings

This section configures default settings for directories on the server and specifies the settings for the root directory. The directory section command can be used extensively to specify particular settings for individual directories.

The following default configuration is specified:

- The `DirectoryIndex` specifies the page to serve to the browser if the folder name only is specified. Traditionally this is the `index.html` file but Windows users may prefer to specify `index.htm` or `default.htm`.
- `IndexOptions` specifies the format for a directory listing where no default html file exists (and the `Indexes` option is specified).
- The `ServerTokens productonly` option limits the information that Apache may present to users.
- The `options` set for the root directory offer locked down security as a default. Individual directories used by the web server need to be specifically set up in the configuration file. For example, the `Options` defaults allow server side includes that can execute operating system commands.
- `AllowOverride None` prevents the use of `.htaccess` files that override security settings.

```
##### DIRECTORY DEFAULTS

<IfModule !mod_access.c>
  # Allows the Order allow,deny commands to work
  LoadModule access_module modules/mod_access.so
</IfModule>

<IfModule !mod_auth.c>
  # Needed for user authorisation (AuthUserFile etc)
  LoadModule auth_module modules/mod_auth.so
</IfModule>

<IfModule !mod_autoindex.c>
  # Allow IndexOptions
  LoadModule autoindex_module modules/mod_autoindex.so
</IfModule>

<IfModule !mod_dir.c>
  # Allow DirectoryIndex
  LoadModule dir_module modules/mod_dir.so
</IfModule>

# .htaccess files can override this configuration for individual folders (if
allowoverride is set).
AccessFileName .htaccess
# Do not allow htaccess files to be read.
<Files ~ "\.ht">
  Order allow,deny
  Deny from all
</Files>

# Default file for a folder.
DirectoryIndex index.html index.htm index.php
```

```

# Default Index listing format.
IndexOptions FancyIndexing

# Server provides limited information on Apache configuration (not OS or loaded
modules)
ServerTokens productonly

# Lock down security on the file system for root folder (i.e. c:\).
<Directory />
  Options None
  AllowOverride None
  Order allow,deny
  Allow from none
</Directory>

```

## Directory Settings

Settings can be easily overridden for individual folders by adding a section for each folder. Note that the path is a physical system path relative to the root of the computer and not relative to the default URL of the server.

The following setting allows access to the web content folder by all users and overrides the locked down access applied to the root folder:

```

##### DIRECTORIES

# Directory options for DocumentRoot
<Directory "c:/web/htdocs">
  Options Indexes IncludesNOEXEC
  AllowOverride None
  Order allow,deny
  Allow from all

  # Custom error page for invalid URL
  # Does not work for Internet Explorer if filesize is too small (less than 512KB).
  ErrorDocument 404 /error404.html

</Directory>

```

The following directory options need to be specified for each directory that is not a subdirectory of the `htdocs` folder for Apache to allow permission to browse the content:

- `allow from all` overrides the default security to allow any user to browse the html content. Try to use only IP addresses if you want to limit access to particular folders.
- `Options Indexes` produces a directory listing if no default html file is found.
- `Options IncludesNOEXEC` allows server side includes but disallows any that attempt to execute an operating system command.
- A number of error types are specified and used by all html browsers allowing the `ErrorDocument` directive to display a web page rather than the default error message.

Many configuration directives can be applied to individual directories. Some common options are:

- `Options ExecCGI` to allow CGI script execution.
- `AllowOverride All` allows an `.htaccess` file to alter the configuration setting individually for that folder.

## Create user directories

User folders allow individual users or departments control over their own web content from a location separate to the main web content.

```
# Allow user directories accessible with http://ServerName/~username
UserDir c:/web/users
```

```
<Directory "c:/web/users">
  Options Indexes IncludesNOEXEC
  Order allow,deny
  Allow from all
</Directory>
```

User folders are easily set up by creating sub-folders in the users folder. The `c:\web\users\marketing` sub-folder contains the marketing department web site accessible using the following URL:

```
http://servername/~marketing
```

## Create a private directory

Private directories require the entry of a username and password before access is given to the web content. Browsing `http://servername/private` prompts for a username and password and allows access only to users that are members of the `private` group as defined in the files that control security.

```
#Private directory with password access for members of private group
```

```
<Directory c:/web/htdocs/private>
  Options Indexes IncludesNOEXEC ExecCGI
  Order allow,deny
  Allow from all
  AuthType BASIC
  AuthName private
  AuthUserFile c:/web/security/users
  AuthGroupFile c:/web/security/groups
  #require valid-user
  require group private
</Directory>
```

The `users` file is a simple colon separated list of usernames and passwords:

```
stamati:crook
fred:bloggs
jane:smith
```

The `groups` file lists the group followed by a colon and a space delimited listing of the users in that group:

```
private: stamati fred
marketing: jane fred
```

The `htpasswd` utility (found in the `bin` folder) can be used to encrypt passwords in the user file. The following command creates the `users` file and adds a user called `stamati` after prompting for a password:

```
htpasswd -c c:\web\security\users stamati
```

Subsequently you can add further users or change existing passwords:

```
htpasswd -b c:\web\security\users stamati password
```

Or delete existing users:

```
htpasswd -D c:\web\security\users stamati
```

Get help as follows:

```
htpasswd -h
```

# Applications

## Activate Server Side Includes

Server Side Includes (SSI) provides a simple way to include a common file within another html file without relying on proprietary mechanisms that may be part of your web design application. A common header or footer file can be easily included into an html file with the following command:

```
<!--#include virtual="test.inc" -->
```

SSI can also display variables such as the current date and time or the modification date of a file:

```
<!--#echo var="DATE_GMT"-->  
<!--#flastmod file="test.inc" -->
```

The mod\_include module is required:

```
# Needed for server side includes (AddOutputFilter INCLUDES)  
LoadModule include_module modules/mod_include.so
```

SSI can then be activated with the following directives for an individual folder:

```
<Directory c:/web/htdocs/test>  
  
Options Indexes IncludesNOEXEC ExecCGI  
Order allow,deny  
Allow from all  
  
# Server side includes for SHTML files  
AddType text/html .shtml  
AddOutputFilter INCLUDES .shtml  
  
</Directory>
```

SSI can be used to CGI programs or execute operating system commands provided that the `Options +Includes` is specified for the required folder. An example command might be:

```
<!--#exec cmd="dir c:\*.*" -->
```

*Note: The `Options +IncludesNOEXEC` is safer to use as it allows files to be included without the ability to call CGI or operating system commands.*

## Install MySQL

Download the Windows binary from <http://www.mysql.com>. If you do not install into the `c:\mysql` folder then you need to copy the `my.cnf` file into a system folder (perhaps `windows/system32`).

Install mysql as a service as follows:

```
mysqld --install
```

Now you can start the service with the service manager or:

```
net start mysql
```

Check the `*.err` file in the `c:\mysql\data` folder if there are any problems.

You now need to change the password for the root super user from the default (mysql) to your own password:

```
c:  
cd\mysql\bin\  
mysqladmin -u root password "newpassword"
```

Set up a more secure system by logging into database as the root user and deleting some of the default users. First log into the mysql system:

```
c:
cd\mysql\bin\
mysql -u root -p
```

You will now be prompted for the password and receive a `mysql>` prompt. MySQL allows any user to access the database from the local machine and you need to delete this facility as follows:

```
DELETE FROM user WHERE Host='localhost' AND User='';
FLUSH PRIVILEGES;
QUIT
```

## **Create a new MySQL database and user**

There are various administration tools for help in administering the MySQL database. Often you just need to create a database with a username and password before installing an application. This is easily done using the MySQL command line.

First log in to the database as the root super user:

```
c:
cd \mysql\bin
mysql -user=root -p
```

After entering the password you can create a database (note the semi-colon to terminate each command):

```
create database phpbb;
```

Now create a new user for that database and assign all privileges:

```
use phpbb;
grant all privileges on phpbb.* to phpbb@localhost identified by 'phpbbpassword';
```

Now quit from the mysql command line and proceed with your installation:

```
quit;
```

## **Backup a mysql database**

The following command is useful to backup the database and will dump the database structure and a series of insert commands to allow the database to be recreated on another machine:

```
c:\mysql\bin\mysqldump databasename > c:\filename.sql
```

Take care with the following command which will install the database on a fresh mysql installation:

```
mysql databasename < c:\filename.sql
```

## **Install PHP**

PHP was designed specifically to provide a programming language for web server applications and many open source packages require that it to be installed alongside Apache. The Windows installation provides limited functionality and so you should download the manual installation zip file from [www.php.net](http://www.php.net):

- Make sure that the MySQL database extensions are included in the distribution.
- Unzip the files and copy into `c:\php`.
- Add `c:\php` into the Windows path.
- Rename the `php.ini-recommended` file to `php.ini`.
- You may need to change the `doc_root` entry in this file to point to the Apache DocumentRoot folder.
- Check that the `php4ts.dll` file is in the `c:\php` folder.

Now add the following into the Apache configuration file:

```
##### PHP CONFIGURATION AND APPLICATIONS
# PHP is loaded as a server module.
# Production servers may be safer configured to use PGP as a CGI function.
```

```
<IfModule !php4apache2.c>
  # For PHP 4 do something like this:
  LoadModule php4_module "c:/php/sapi/php4apache2.dll"
  AddType application/x-httpd-php .php
</IfModule>
```

```
#<IfModule !php5apache2.c>
# For PHP 5 do something like this:
# LoadModule php5_module "c:/php/php5apache2.dll"
# AddType application/x-httpd-php .php
#</IfModule>
```

```
# configure the path to php.ini
PHPIniDir "C:/php"
```

You can test by creating an html file with a `.php` extension and add the following into the html:

```
<p><?phpinfo()?></p>
```

Information on security and configuring for CGI is detailed in the `c:\php\install.txt` document.

*Note: The PHP website warns that PHP is not suitable for use with Apache version 2.0 due to problems with some PHP modules with threading. They recommend that Apache 1.3.x is used with PHP, however this does not run well on Windows.*

## Install Web Applications

The hard work is now complete and you are in a position to install a wide range of web applications that conform to the LAMP (Linux-Apache-MySQL-PHP) paradigm.

Installation typically involves the following stages:

- Copy the php files to a new folder.
- Set up security and add an `alias` to point the new folder.
- Create a new MySQL database and user.
- Configure any special aspects of your php installation.
- Browse the `index.php` page which will prompt you through the database and application installation.
- Delete the installation folders.

We recommend the following open source PHP software:

- `phpBB` is the leading open source bulletin board and installs without any problems.
- `MyPHPAdmin` is a web application for administering MySQL.
- `Mambo` is a state of the art portal application.
- `Oscommerce` is a leading open source shopping cart.

The PHP configuration file (`php.ini`) may need changing to accommodate the requirements of each application. An alternative is to use Apache directives to override the default settings. The `php_admin_flag` and `php_admin_value` directives need to be set for the whole server whilst the `php_flag` and `php_value` settings can be made at directory level.

*Note: Remember that `c:\php` must be in the path.*

The following configuration file is sufficient to allow the default PHP 4 installation to function with the listed open source web applications (all installed as sub-folder of the `c:\web\apps` folder):

```
<Directory "c:/web/apps">
Options Indexes IncludesNOEXEC ExecCGI
  AllowOverride None
Order allow,deny
  Allow from all
```

```

</Directory>

# Essential for Mamabo.
php_admin_value session.save_path c:\temp

# Recommended for Mamabo.
php_admin_flag display_errors On
php_admin_flag magic_quotes_gpc On

# Essential for oscommerce
php_admin_flag register_globals On

# PORTAL - MAMBO
Alias /portal "c:/web/apps/mambo-4.5.1"
<Directory "c:/web/phpapps/mambo-4.5.1">
</Directory>

# FORUM - PHPBB
Alias /forum "c:/web/apps/phpbb2"

# SHOP - OSCOMMERCE
Alias /shop "C:/web/apps/oscommerce-2.2ms2/catalog"
<Directory "C:/web/apps/oscommerce-2.2ms2">
</Directory>

# PHPMYADMIN
Alias /phpmyadmin "C:/web/apps/phpMyAdmin-2.6.1-rc1"

```

## Security

Windows security is not covered in this document. If you are running a production server you will need firewall protection and the latest security updates on your machine.

Windows services should be configured to run under user accounts with restricted security. This applies to the MySQL service as well as Apache. The accounts need to have the `log on as a service` option and provide limited access to the machine and the network. The Apache service requires the following directory access:

- RX on the web content folders.
- RWXD on the log folders and files.
- RX on the Apache executables.

## Linux Notes

Most of this document also applies to a Linux installation and it is easy to move your web installation to Linux if required. The following brief notes assume that you have installed Fedora 2 core Linux and selected the server option which pre-installs the Apache web server for you.

*Note: The location of files and facilities available will vary according to the flavour of Linux.*

Installing Apache using the rpm installer puts the files into the following folders:

- `/etc/httpd/` for the `bin` executables and `conf`, `logs` and `modules` folders. The Apache executable is called `httpd`.
- `/var/www/html` for html content.
- `/var/log/httpd` for log files.
- `/var/logs/httpd` for log files.
- `/usr/sbin` and `/usr/bin` for the utility programs such as `apachectl` and `htpasswd`.

*Note: Compiling Apache from source normally places all files into the `/usr/local/apache` folder.*

The `apachectl` program is used to start and stop Apache:

```
apachectl -k start
apachectl -k stop
```

The configuration file and directives should already be familiar and require UNIX paths to the relevant folders. The locations of files may vary according to your installation. The `mime.types` file for example may be located in the `/etc/` folder instead of `conf`.

Apache for Linux uses a different core processing module and needs the following performance directives:

```
<IfModule prefork.c>
    StartServers      8
    MinSpareServers   5
    MaxSpareServers   20
    MaxClients        150
    MaxRequestsPerChild 4000
</IfModule>
```

```
# worker MPM
<IfModule worker.c>
    StartServers      2
    MaxClients        150
    MinSpareThreads   25
    MaxSpareThreads   75
    ThreadsPerChild   25
    MaxRequestsPerChild 0
</IfModule>
```

Apache also requires a user and group to be specified in the configuration file. The Apache user may have been created on installation and you can assign restrictive permissions to this account and group if required. The username and group must be documented in the configuration file:

```
user apache
group apache
```

Apache should normally be set up to run at run levels 3 and above. This means that the web server will run as soon as Linux enters multi-user networking mode even if the graphics front end is not running. Check the configuration as follows:

```
chkconfig --list httpd
```

If the service is not running as required then configure it to run automatically from run levels 3 and above:

```
chkconfig --level 345 httpd on
```

You can check that the service is set up as follows:

```
/etc/rc.d/init.d/httpd status
```

Use the following command to shutdown your server. You can use the following command to shut down your server gracefully:

```
shutdown -v now
```

## Conclusion

Apache is the industry standard web server and has many features used for very large and very busy web sites. The **redware** configuration focuses on creating a comprehensible configuration of a simple web server suitable for individual or small business use.

Some of the features that Apache can provide include:

- Very sophisticated features for hosting multiple web sites from a single server.
- Adding multiple languages to your web site (see the default installation configuration).

- Allowing SSL security for communication with the `https:` protocol.
- Implementation of a proxy server with disk caching to allow multiple browsers on an internal network browser access to the internet via a single server. This avoids the requirement to set up network address translation when connecting the internal network to the internet.
- Ability to create a reverse proxy to provide secure relay of web pages from multiple internal servers to the internet.
- Integration with perl or python for running web applications.
- Integration with tomcat for running Java servlet (jsp) web applications.
- Ability to load balance between servers for very busy sites.
- Security defined by IP address.
- The `rewrite` module which can rewrite incoming URLs to cope with a variety of network configurations.
- And so forth...

## References

Apache Documentation. Available with your installation or at <http://www.apache.org>.

Apache. The definitive guide. 3<sup>rd</sup> Edition. Laurie B. and Laurie P. O'Reilly 2002. ISBN 0-596-00203-3.

## httpd.conf

The complete file is listed below and can be used to replace the default `httpd.conf` file in the `conf` sub-folder of the apache installation.

The following directory structure is assumed (and can be downloaded from the `web.zip` file on our download page):

- Apache installation folder is `c:\program files\apache group\apache2`.
- Web content is available in `c:\web\htdocs`.
- User sub-folders are set up in `c:\web\users` if required,
- Private web content is set up in `c:\web\htdocs\private`.
- Username and passwords are set up in the `users` and `groups` files of the `c:\web\security` folder.

Remember to change the folder names as required in the configuration file as well as the `ServerName` and `ServerAdmin` settings:

```
# bof httpd.conf
# www.redware.com 28 January 2005 Version 1.6

# redware Apache 2.0.x configuration file for standard Windows installation
# on C: drive with document folder in c:/web/htdocs.

###TODO: Change these entries for standard Apache installation.
###TODO: Change references to c:/ and c:/web as required.

##### MINIMAL
# Listen out for browsers on port 80.
Listen 80

# ServerName and Port used to create redirection URLs.
# Should be a valid DNS name or the IP address if there is no DNS.

###TODO: Change to the machine IP address.
ServerName 10.0.1.4:80

# Location for conf, error, and log folders (no trailing slash).
ServerRoot "c:/Program Files/Apache Group/Apache2"
```

```

# Location of html pages (no trailing slash)
DocumentRoot "c:/web/htdocs"

# Administrator's email
###TODO: Change to administrator email.
ServerAdmin website@domain.com

##### PERFORMANCE

# http network level performance settings
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

# WinNT MPM - specific windows recommended settings.
# ThreadsPerChild: constant number of worker threads in the server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_winnt.c>
    ThreadsPerChild 50
    MaxRequestsPerChild 0
</IfModule>

# Off stops apache substituting ServerName and Port for redirection.
UseCanonicalName Off

# Unreliable way to get user name from the browser so not used.
IdentityCheck off

# Make sure DNS lookups are disabled (this is the default).
HostnameLookups Off

# Performance notes:
# Set MaxClients if your server has too many users.
# Add the browsermatch directives from the default configuration.
# Different settings required for Linux.

##### MIME TYPES

<IfModule !mod_mime.c>
    # Allows TypesConfig to work
    LoadModule mime_module modules/mod_mime.so
</IfModule>

DefaultType text/plain

# Map files to mime types
# Required for Mozilla Firefox to display html files as html and not text.
TypesConfig conf/mime.types

# Add the mime_magic module if more mime types are needed.
#<IfModule mod_mime_magic.c>
#    MIMEMagicFile conf/magic
#</IfModule>

```

```

##### LOGS

<IfModule !mod_log_config.c>
  # Needed for LogFormat
  LoadModule log_config_module modules/mod_log_config.so
</IfModule>

# Errorlog (relative to ServerRoot).
ErrorLog logs/error.log

#PidFile not really useful for Windows - used to kill process in UNIX.
PidFile logs/http.pid

# Common Log format (not used).
#Log File in the Common Log file format.
#LogFormat "%h %l %u %t \"%r\" %>s %b" common
#CustomLog logs/access.log common

# Combined log format.
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
#CustomLog logs/access.log combined

# Optionally rotate logs to a new file every 24 hours.
CustomLog "|bin/rotatelogs.exe logs/log%Y%m%d%H.log 86400" combined

##### REDIRECTS and ALIAS

<IfModule !mod_alias.c>
  # Needed for redirect and alias
  LoadModule alias_module modules/mod_alias.so
</IfModule>

# Redirection for the apache manual
Redirect /manual http://httpd.apache.org/docs-2.0/

##### DIRECTORY DEFAULTS

<IfModule !mod_access.c>
  # Allows the Order allow,deny commands to work
  LoadModule access_module modules/mod_access.so
</IfModule>

<IfModule !mod_auth.c>
  # Needed for user authorisation (AuthUserFile etc)
  LoadModule auth_module modules/mod_auth.so
</IfModule>

<IfModule !mod_autoindex.c>
  # Allow IndexOptions
  LoadModule autoindex_module modules/mod_autoindex.so
</IfModule>

<IfModule !mod_dir.c>

```

```

# Allow DirectoryIndex
LoadModule dir_module modules/mod_dir.so
</IfModule>

# .htaccess files can override this configuration for individual folders (if
allowoverride is set).
AccessFileName .htaccess
# Do not allow htaccess files to be read.
<Files ~ "\.ht">
    Order allow,deny
    Deny from all
</Files>

# Default file for a folder.
DirectoryIndex index.html index.htm index.php

# Default Index listing format.
IndexOptions FancyIndexing

# Server provides limited information on Apache configuration (not OS or loaded
modules)
ServerTokens productonly

# Lock down security on the file system for root folder (i.e. c:\).
<Directory />
    Options None
    AllowOverride None
    Order allow,deny
    Allow from none
</Directory>

##### DIRECTORIES

# Directory options for DocumentRoot
<Directory "c:/web/htdocs">
    Options Indexes IncludesNOEXEC
    AllowOverride None
    Order allow,deny
    Allow from all

# Custom error page for invalid URL
# Does not work for Internet Explorer if filesize is too small (less than 512KB).
ErrorDocument 404 /error404.html

</Directory>

#Private directory with password access for members of private group
<Directory c:/web/htdocs/private>
    Options Indexes IncludesNOEXEC
    Order allow,deny
    Allow from all
    AuthType BASIC
    AuthName private
    AuthUserFile c:/web/security/users
    AuthGroupFile c:/web/security/groups

```

```
#require valid-user
require group private
</Directory>

##### USER DIRECTORIES

<IfModule !mod_userdir.c>
  # Needed for user directories
  LoadModule userdir_module modules/mod_userdir.so
</IfModule>

# Allow user directories accessible with http://ServerName/~username
UserDir c:/web/users

# Disable the root user directory on UNIX.
UserDir root disable

<Directory "c:/web/users">
  Options Indexes IncludesNOEXEC ExecCGI
  Order allow,deny
  Allow from all
</Directory>

##### ADDITIONAL CONFIGURATION FILES
# Include redware configuration file.
# Include conf/redware.conf

# eof httpd.conf
```