Using Apache with Microsoft Windows

This document explains how to install, configure and run Apache 2.0 under Microsoft Windows. If you find any bugs, or wish to contribute in other ways, please use our bug reporting page[1].

This document assumes that you are installing a binary distribution of Apache. If you want to compile Apache yourself (possibly to help with development or tracking down bugs), see Compiling Apache for Microsoft Windows[2].

Because of the current versioning policies on Microsoft Windows operating system families, this document assumes the following:

- **Windows NT:** This means all versions of Windows that are based on the Windows NT kernel. Includes Windows NT, Windows 2000, Windows XP and Windows .Net Server 2003.
- **Windows 9x:** This means older, consumer-oriented versions of Windows. Includes Windows 95 (also OSR2), Windows 98 and Windows ME.

Topics

Operating System Requirements

The primary Windows platform for running Apache 2.0 is Windows NT. The binary installer only works with the x86 family of processors, such as Intel and AMD processors. Running Apache on Windows 9x is not thoroughly tested, and it is never recommended on production systems.

On all operating systems, TCP/IP networking must be installed and working. If running on Windows 95, the Winsock 2 upgrade must be installed. Winsock 2 for Windows 95 can be downloaded from here[3].

On Windows NT 4.0, installing Service Pack 6 is strongly recommended, as Service Pack 4 created known issues with TCP/IP and Winsock integrity that were resolved in later Service Packs.

Downloading Apache for Windows

Information on the latest versions of Apache can be found on the web site of the Apache web server at http://httpd.apache.org/download.cgi. There you will find the current release, as well as more recent alpha or beta test versions, and a list of HTTP and FTP mirrors from which you can download the Apache web server. Please use a mirror near to you for a fast and reliable download.

For Windows installations you should download the version of Apache for Windows with the `.msi` extension. This is a single Microsoft Installer file, which contains a ready-to-run version of Apache. There is a separate `.zip` file, which contains only the source code. You can compile Apache yourself with the Microsoft Visual C++ (Visual Studio) tools.

Installing Apache for Windows

You need Microsoft Installer 1.2 or above for the installation to work. On Windows 9x you can update your Microsoft Installer to version 2.0 here[4] and on Windows NT 4.0 and 2000 the version 2.0 update can be found here[5]. Windows XP does not need this update.

Note that you cannot install two versions of Apache 2.0 on the same computer with the binary installer. You can, however, install a version of the 1.3 series **and** a version of the 2.0 series on the same computer without problems. If you need to have two different 2.0 versions on the same computer, you have to compile and install Apache from the source[2].

Run the Apache `.msi` file you downloaded above. The installation will ask you for these things:

1. **Network Domain.** Enter the DNS domain in which your server is or will be registered in. For example, if your server's full DNS name is `server.mydomain.net`, you would type `mydomain.net` here.
2. **Server Name.** Your server's full DNS name. From the example above, you would type `server.mydomain.net` here.
3. **Administrator's Email Address.** Enter the server administrator's or webmaster's email address here. This address will be displayed along with error messages to the client by default.
4. **For whom to install Apache** Select `for All Users, on Port 80, as a Service - Recommended` if you'd like your new Apache to listen at port 80 for incoming traffic. It will run as a service (that is, Apache will run even if no one is logged in on the server at the moment) Select `only for the Current User, on Port 8080, when started Manually` if you'd like to install Apache for your personal experimenting or if you already have another WWW server running on port 80.
5. **The installation type.** Select `Typical` for everything except the source code and libraries for module development. With `Custom` you can specify what to install. A full install will require about 13 megabytes of free disk space. This does *not* include the size of your web site(s).
6. **Where to install.** The default path is `C:\Program Files\Apache Group` under which a directory called `Apache2` will be created by default.

During the installation, Apache will configure the files in the `conf` subdirectory to reflect the chosen installation directory. However, if any of the configuration files in this directory already exist, they will not be overwritten. Instead, the new copy of the corresponding file will be left with the extension `.default`. So, for example, if `conf\httpd.conf` already exists, it will be renamed as `conf\httpd.conf.default`. After the installation you should manually check to see what new settings are in the `.default` file, and if necessary, update your existing configuration file.

Also, if you already have a file called `htdocs\index.html`, it will not be overwritten (and no `index.html.default` will be installed either). This means it should be safe to install Apache over an existing installation, although you would have to stop the existing running server before doing the installation, and then start the new one after the installation is finished.

After installing Apache, you must edit the configuration files in the `conf` subdirectory as required. These files will be configured during the installation so that Apache is ready to be run from the directory it was installed into, with the documents server from the subdirectory `htdocs`. There are lots of other options which you should set before you really start using Apache. However, to get started quickly, the files should work as installed.

Customizing Apache for Windows

Apache is configured by the files in the `conf` subdirectory. These are the same files used to configure the Unix version, but there are a few different directives for Apache on Windows. See the directive index[6] for all the available directives.

The main differences in Apache for Windows are:

- Because Apache for Windows is multithreaded, it does not use a separate process for each request, as Apache does on Unix. Instead there are usually only two Apache processes running: a parent process, and a child which handles the requests. Within the child process each request is handled by a separate thread.

  The process management directives are also different:

  MaxRequestsPerChild: Like the Unix directive, this controls how many requests a single child process will serve before exiting. However, unlike on Unix, a single process serves all the requests at once, not just one. If this is set, it is recommended that a very high number is used. The recommended default, MaxRequestsPerChild 0, causes the child process to never exit.

  > **Warning: The server configuration file is reread when a new child process is started. If you have modified httpd.conf, the new child may not start or you may receive unexpected results.**

  ThreadsPerChild: This directive is new. It tells the server how many threads it should use. This is the maximum number of connections the server can handle at once, so be sure to set this number high enough for your site if you get a lot of hits. The recommended default is ThreadsPerChild 50.

- The directives that accept filenames as arguments must use Windows filenames instead of Unix ones. However, because Apache uses Unix-style names internally, you must use forward slashes, not backslashes. Drive letters can be used; if omitted, the drive with the Apache executable will be assumed.

- Apache for Windows contains the ability to load modules at runtime, without recompiling the server. If Apache is compiled normally, it will install a number of optional modules in the \Apache2\modules directory. To activate these or other modules, the new LoadModule directive must be used. For example, to activate the status module, use the following (in addition to the status-activating directives in access.conf):

  ```
  LoadModule status_module modules/mod_status.so
  ```

  Information on creating loadable modules[7] is also available.

- Apache can also load ISAPI (Internet Server Application Programming Interface) extensions (i.e. internet server applications), such as those used by Microsoft IIS and other Windows servers. More information is available[8]. Note that Apache **cannot** load ISAPI Filters.

- When running CGI scripts, the method Apache uses to find the interpreter for the script is configurable using the ScriptInterpreterSource directive.

- Since it is often difficult to manage files with names like .htaccess in Windows, you may find it useful to change the name of this per-directory configuration file using the AccessFilename directive.

- Any errors during Apache startup are logged into the Windows event log when running on Windows NT. This mechanism acts as a backup for those situations where Apache cannot even access the normally used error.log file. You can view the Windows event log by using the Event Viewer application on Windows NT 4.0, and the Event Viewer MMC snap-in on newer versions of Windows.

Note that there is no startup error logging on Windows 9x because no Windows event log exists on those operating systems.

Running Apache as a Service

Apache can be run as a service on Windows NT. There is some highly experimental support for similar behavior on Windows 9x.

You can install Apache as a service automatically during the installation. If you chose to install for all users, the installation will create an Apache service for you. If you specify to install for yourself only, you can manually register Apache as a service after the installation. You have to be a member of the Administrators group for the service installation to succeed.

Apache comes with a utility called the Apache Service Monitor. With it you can see and manage the state of all installed Apache services on any machine on your network. To be able to manage an Apache service with the monitor, you have to first install the service (either automatically via the installation or manually).

You can install Apache as a Windows NT service as follows from the command prompt at the Apache `bin` subdirectory:

```
apache -k install
```

If you need to specify the name of the service you want to install, use the following command. You have to do this if you have several different service installations of Apache on your computer.

```
apache -k install -n "MyServiceName"
```

If you need to have specifically named configuration files for different services, you must use this:

```
apache -k install -n "MyServiceName" -f "c:\files\my.conf"
```

If you use the first command without any special parameters except `-k install`, the service will be called `Apache2` and the configuration will be assumed to be `conf\httpd.conf`.

Removing an Apache service is easy. Just use:

```
apache -k uninstall
```

The specific Apache service to be uninstalled can be specified by using:

```
apache -k uninstall -n "MyServiceName"
```

Normal starting, restarting and shutting down of an Apache service is usually done via the Apache Service Monitor, by using commands like `NET START Apache2` and `NET STOP Apache2` or via normal Windows service management. Before starting Apache as a service by any means, you should test the service's configuration file by using:

```
apache -n "MyServiceName" -t
```

You can control an Apache service by its command line switches, too. To start an installed Apache service you'll use this:

```
apache -k start
```

To stop an Apache service via the command line switches, use this:

```
apache -k stop
```

or

```
apache -k shutdown
```

You can also restart a running service and force it to reread its configuration file by using:

```
apache -k restart
```

By default, all Apache services are registered to run as the system user (the `LocalSystem` account). The `LocalSystem` account has no privileges to your network via any Windows-secured mechanism, including the file system, named pipes, DCOM, or secure RPC. It has, however, wide privileges locally.

Never grant any network privileges to the `LocalSystem` account! If you need Apache to be able to access network resources, create a separate account for Apache as noted below.

You may want to create a separate account for running Apache service(s). Especially, if you have to access network resources via Apache, this is strongly recommended.

1. Create a normal domain user account, and be sure to memorize its password.
2. Grant the newly-created user a privilege of `Log on as a service` and `Act as part of the operating system`. On Windows NT 4.0 these privileges are granted via User Manager for Domains, but on Windows 2000 and XP you probably want to use Group Policy for propagating these settings. You can also manually set these via the Local Security Policy MMC snap-in.
3. Confirm that the created account is a member of the Users group.
4. Grant the account read and execute (RX) rights to all document and script folders (`htdocs` and `cgi-bin` for example).
5. Grant the account change (RWXD) rights to the Apache `logs` directory.
6. Grant the account read and execute (RX) rights to the `Apache.exe` binary executable.

It is usually a good practice to grant the user the Apache service runs as read and execute (RX) access to the whole Apache2 directory, except the `logs` subdirectory, where the user has to have at least change (RWXD) rights.

If you allow the account to log in as a user and as a service, then you can log on with that account and test that the account has the privileges to execute the scripts, read the web pages, and that you can start Apache in a console window. If this works, and you have followed the steps above, Apache should execute as a service with no problems.

Error code 2186 is a good indication that you need to review the "Log On As" configuration for the service, since Apache cannot access a required network resource. Also, pay close attention to the privileges of the user Apache is configured to run as.

When starting Apache as a service you may encounter an error message from the Windows Service Control Manager. For example, if you try to start Apache by using the Services applet in the Windows Control Panel, you may get the following message:

```
Could not start the Apache2 service on \\COMPUTER
Error 1067; The process terminated unexpectedly.
```

You will get this generic error if there is any problem with starting the Apache service. In order to see what is really causing the problem you should follow the instructions for Running Apache for Windows from the Command Prompt.

There is some support for Apache on Windows 9x to behave in a similar manner as a service on Windows NT. It is **highly experimental**. It is not of production-class reliability, and its future is not guaranteed. It can be mostly regarded as a risky thing to play with - proceed with caution!

There are some differences between the two kinds of services you should be aware of:

- Apache will attempt to start and if successful it will run in the background. If you run the command

  ```
  apache -n "MyServiceName" -k start
  ```

  via a shortcut on your desktop, for example, then if the service starts successfully, a console window will flash up but it immediately disappears. If Apache detects any errors on startup such as incorrect entries in the httpd.conf configuration file, the console window will remain visible. This will display an error message which will be useful in tracking down the cause of the problem.
- Windows 9x does not support `NET START` or `NET STOP` commands. You must control the Apache service on the command prompt via the `-k` switches.
- Apache and Windows 9x offer no support for running Apache as a specific user with network privileges. In fact, Windows 9x offers no security on the local machine, either. This is the simple reason because of which the Apache Software Foundation never endorses use of a Windows 9x -based system as a public Apache server. The primitive support for Windows 9x exists only to assist the user in developing web content and learning the Apache server, and perhaps as an intranet server on a secured, private network.

Once you have confirmed that Apache runs correctly as a console application you can install, control and uninstall the pseudo-service with the same commands as on Windows NT. You can also use the Apache Service Monitor to manage Windows 9x pseudo-services.

Running Apache as a Console Application

Running Apache as a service is usually the recommended way to use it, but it is sometimes easier to work from the command line (on Windows 9x running Apache from the command line is the recommended way due to the lack of reliable service support.)

To run Apache from the command line as a console application, use the following command:

```
apache
```

Apache will execute, and will remain running until it is stopped by pressing Control-C.

You can also run Apache via the shortcut Start Apache in Console placed to `Start Menu --> Programs --> Apache HTTP Server 2.0.xx --> Control Apache Server` during the installation. This will open a console window and start Apache inside it. If you don't have Apache installed as a service, the window will remain visible until you stop Apache by pressing Control-C in the console window where Apache is running in. The server will exit in a few seconds. However, if you do have Apache installed as a service, the shortcut starts the service. If the Apache service is running already, the shortcut doesn't do anything.

You can tell a running Apache to stop by opening another console window and entering:

```
apache -k shutdown
```

This should be preferred over pressing Control-C because this lets Apache end any current operations and clean up gracefully.

You can also tell Apache to restart. This forces it to reread the configuration file. Any operations in progress are allowed to complete without interruption. To restart Apache, use:

```
apache -k restart
```

> Note for people familiar with the Unix version of Apache: these commands provide a Windows equivalent to `kill -TERM` *`pid`* and `kill -USR1` *`pid`*. The command line option used, `-k`, was chosen as a reminder of the `kill` command used on Unix.

If the Apache console window closes immediately or unexpectedly after startup, open the Command Prompt from the Start Menu --> Programs. Change to the folder to which you installed Apache, type the command `apache`, and read the error message. Then change to the logs folder, and review the `error.log` file for configuration mistakes. If you accepted the defaults when you installed Apache, the commands would be:

```
c:
cd "\Program Files\Apache Group\Apache2\bin"
apache
```

Then wait for Apache to stop, or press Control-C. Then enter the following:

```
cd ..\logs
more < error.log
```

When working with Apache it is important to know how it will find the configuration file. You can specify a configuration file on the command line in two ways:

- `-f` specifies an absolute or relative path to a particular configuration file:

  ```
  apache -f "c:\my server files\anotherconfig.conf"
  ```

  or

```
apache -f files\anotherconfig.conf
```

- `-n` specifies the installed Apache service whose configuration file is to be used:

```
apache -n "MyServiceName"
```

In both of these cases, the proper `ServerRoot` should be set in the configuration file.

If you don't specify a configuration file with `-f` or `-n`, Apache will use the file name compiled into the server, such as `conf\httpd.conf`. This built-in path is relative to the installation directory. You can verify the compiled file name from a value labelled as `SERVER_CONFIG_FILE` when invoking Apache with the `-V` switch, like this:

```
apache -V
```

Apache will then try to determine its `ServerRoot` by trying the following, in this order:

1. A `ServerRoot` directive via the `-C` command line switch.
2. The `-d` switch on the command line.
3. Current working directory.
4. A registry entry which was created if you did a binary installation.
5. The server root compiled into the server. This is `/apache` by default, you can verify it by using `apache -V` and looking for a value labelled as `HTTPD_ROOT`.

During the installation, a version-specific registry key is created in the Windows registry. The location of this key depends on the type of the installation. If you chose to install Apache for all users, the key is located under the `HKEY_LOCAL_MACHINE` hive, like this (the version numbers will of course vary between different versions of Apache:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Apache Group\Apache\2.0.43
```

Correspondingly, if you chose to install Apache for the current user only, the key is located under the `HKEY_CURRENT_USER` hive, the contents of which are dependent of the user currently logged on:

```
HKEY_CURRENT_USER\SOFTWARE\Apache Group\Apache\2.0.43
```

This key is compiled into the server and can enable you to test new versions without affecting the current version. Of course, you must take care not to install the new version in the same directory as another version.

If you did not do a binary install, Apache will in some scenarios complain about the missing registry key. This warning can be ignored if the server was otherwise able to find its configuration file.

The value of this key is the `ServerRoot` directory which contains the `conf` subdirectory. When Apache starts it reads the `httpd.conf` file from that directory. If this file contains a `ServerRoot` directive which contains a different directory from the one obtained from the registry key above, Apache will forget the registry key and use the directory from the configuration file. If you copy the Apache directory or configuration files to a new location it is vital that you update the `ServerRoot` directive in the `httpd.conf` file to reflect the new location.

Testing the Installation

After starting Apache (either in a console window or as a service) it will be listening on port 80 (unless you changed the `Listen` directive in the configuration files or installed Apache only for the current user). To connect to the server and access the default page, launch a browser and enter this URL:

```
http://localhost/
```

Apache should respond with a welcome page and a link to the Apache manual. If nothing happens or you get an error, look in the `error.log` file in the `logs` subdirectory. If your host is not connected to the net, or if you have serious problems with your DNS (Domain Name Service) configuration, you may have to use this URL:

```
http://127.0.0.1/
```

Once your basic installation is working, you should configure it properly by editing the files in the `conf` subdirectory. Again, if you change the configuration of the Windows NT service for Apache, first attempt to start it from the command line to make sure that the service starts with no errors.

Because Apache **cannot** share the same port with another TCP/IP application, you may need to stop, uninstall or reconfigure certain other services before running Apache. These conflicting services include other WWW servers and some firewall implementations.

URI References

[1] http://httpd.apache.org/bug_report.html

[2] http://httpd.apache.org/docs-2.1/platform/win_compiling.html

[3] http://www.microsoft.com/windows95/downloads/contents/WUAdminTools/S_WUNetworkingTools/W95Sockets2/Default.asp

[4] http://www.microsoft.com/downloads/release.asp?ReleaseID=32831

[5] http://www.microsoft.com/downloads/release.asp?ReleaseID=32832

[6] http://httpd.apache.org/docs-2.1/mod/directives.html

[7] http://httpd.apache.org/docs-2.1/mod/mod_so.html#creating

[8] http://httpd.apache.org/docs-2.1/mod/mod_isapi.html