

Compiling Apache for Microsoft Windows

There are many important points before you begin compiling Apache. See [Using Apache with Microsoft Windows](#)¹ before you begin.

Topics

Requirements	1
Command-Line Build	2
Developer Studio Workspace IDE Build.....	3
Project Components	3
URI References	5

Requirements

Compiling Apache requires the following environment to be properly installed:

- Disk Space

Make sure you have at least 50 MB of free disk space available. After installation Apache requires approximately 10 MB of disk space, plus space for log and cache files, which can grow rapidly. The actual disk space requirements will vary considerably based on your chosen configuration and any third-party modules or libraries.

- Microsoft Visual C++ 5.0 or higher.

Apache can be built using the command line tools, or from within the Visual Studio IDE Workbench. The command line build requires the environment to reflect the `PATH`, `INCLUDE`, `LIB` and other variables that can be configured with the `vcvars32` batch file:

```
"c:\Program Files\DevStudio\VC\Bin\vcvars32.bat"
```

- The Windows Platform SDK.

Visual C++ 5.0 builds require an updated Microsoft Windows Platform SDK to enable some Apache features. For command line builds, the Platform SDK environment is prepared by the `setenv` batch file:

```
"c:\Program Files\Platform SDK\setenv.bat"
```

The Platform SDK files distributed with Visual C++ 6.0 and later are sufficient, so users of later version may skip this requirement.

Note that the Windows Platform SDK update is required to enable all supported `mod_isapi` features. Without a recent update, Apache will issue warnings under MSVC++ 5.0 that some `mod_isapi` features will be disabled. Look for the update at <http://msdn.microsoft.com/downloads/sdks/platform/platform.asp>.

- The awk utility (awk, gawk or similar).

To install Apache within the build system, several files are modified using the `awk.exe` utility. `awk` was chosen since it is a very small download (compared with Perl or WSH/VB) and accomplishes the task of generating files. Brian Kernighan's <http://cm.bell-labs.com/cm/cs/who/bwk/> site has a compiled native Win32 binary, <http://cm.bell-labs.com/cm/cs/who/bwk/awk95.exe> which you must save with the name `awk.exe` rather than `awk95.exe`.

Compiling Apache for Microsoft Windows

Note that Developer Studio IDE will only find `awk.exe` from the Tools menu Options... Directories tab (the Projects - VC++ Directories pane in Developer Studio 7.0) listing Executable file paths. Add the path for `awk.exe` to this list, and your system `PATH` environment variable, as needed.

Also note that if you are using Cygwin (<http://www.cygwin.com/>) the `awk` utility is named `gawk.exe` and that the file `awk.exe` is really a symlink to the `gawk.exe` file. The Windows command shell does not recognize symlinks, and because of that building `InstallBin` will fail. A workaround is to delete `awk.exe` from the cygwin installation and rename `gawk.exe` to `awk.exe`.

- [Optional] OpenSSL libraries (for `mod_ssl` and `ab.exe` with ssl support)

Caution: there are significant restrictions and prohibitions on the use and distribution of strong cryptography and patented intellectual property throughout the world. OpenSSL includes strong cryptography controlled by both export regulations and domestic law, as well as intellectual property protected by patent, in the United States and elsewhere. Neither the Apache Software Foundation nor the OpenSSL project can provide legal advise regarding possession, use, or distribution of the code provided by the OpenSSL project. **Consult your own legal counsel, you are responsible for your own actions.**

OpenSSL must be installed into a `srclib` subdirectory named `openssl`, obtained from <http://www.openssl.org/source/>, in order to compile `mod_ssl` or the `abs` project (`ab.exe` with SSL support.) To prepare OpenSSL for both `release` and `debug` builds of Apache, and disable the patent protected features in 0.9.6, you might use the following build commands:

```
perl util\mkfiles.pl >MININFO
perl util\mk1mf.pl dll no-asm no-mdc2 no-rc5 no-idea VC-WIN32
>makefile
perl util\mk1mf.pl dll debug no-asm no-mdc2 no-rc5 no-idea VC-WIN32
>makefile.dbg
perl util\mkdef.pl 32 libeay no-asm no-mdc2 no-rc5 no-idea
>ms\libeay32.def
perl util\mkdef.pl 32 ssleay no-asm no-mdc2 no-rc5 no-idea
>ms\ssleay32.def
nmake
nmake -f makefile.dbg
```

- [Optional] zlib sources (for `mod_deflate`)

Zlib must be installed into a `srclib` subdirectory named `zlib`, however those sources need not be compiled. The build system will compile the compression sources directly into the `mod_deflate` module. Zlib can be obtained from <http://www.gzip.org/zlib/> -- `mod_deflate` is confirmed to build correctly with version 1.1.4.

Command-Line Build

First, unpack the Apache distribution into an appropriate directory. Open a command-line prompt and `cd` to that directory.

The master Apache makefile instructions are contained in the `Makefile.win` file. To compile Apache on Windows NT, simply use one of the following commands to compiled the `release` or `debug` build, respectively:

Compiling Apache for Microsoft Windows

```
nmake /f Makefile.win _apacher  
nmake /f Makefile.win _apached
```

Either command will compile Apache. The latter will include debugging information in the resulting files, making it easier to find bugs and track down problems.

Developer Studio Workspace IDE Build

Apache can also be compiled using VC++'s Visual Studio development environment. To simplify this process, a Visual Studio workspace, `Apache.dsw`, is provided. This workspace exposes the entire list of working `.dsp` projects that are required for the complete Apache binary release. It includes dependencies between the projects to assure that they are built in the appropriate order.

Open the `Apache.dsw` workspace, and select `InstallBin` (Release or Debug build, as desired) as the Active Project. `InstallBin` causes all related project to be built, and then invokes `Makefile.win` to move the compiled executables and dlls. You may personalize the `INSTDIR`=choice by changing `InstallBin`'s Settings, General tab, Build command line entry. `INSTDIR` defaults to the `/Apache2` directory. If you only want a test compile (without installing) you may build the `BuildBin` project instead.

The `.dsp` project files are distributed in Visual C++ 6.0 format. Visual C++ 5.0 (97) will recognize them. Visual C++ 7.0 (.net) must convert `Apache.dsw` plus the `.dsp` files into an `Apache.sln` plus `.msproj` files, be sure you reconvert the `.msproj` file if any of the source `.dsp` files change! This is really trivial, just open `Apache.dsw` in the VC++ 7.0 IDE once again.

Visual C++ 7.0 (.net) users should also use the Build menu, Configuration Manager dialog to uncheck both the Debug and Release Solution modules `abs`, `mod_ssl` and `mod_deflate`. These modules are built by invoking `nmake` or the IDE directly with the `BinBuild` target to build those modules explicitly, only if the `srclib` directories `openssl` and/or `zlib` exist.

Exported `.mak` files pose a greater hassle, but they are required for Visual C++ 5.0 users to build `mod_ssl`, `abs` (ab with SSL support) and/or `mod_deflate`. VC++ 7.0 (.net) users also benefit, `nmake` builds are faster than `binenv` builds. Build the entire project from within the VC++ 5.0 or 6.0 IDE, then use the Project Menu Export for all makefiles. You must build the projects first in order to create all dynamic auto-generated targets, so that dependencies can be parsed correctly. Run the following command to fix the paths so they will build anywhere:

```
perl srclib\apr\build\fixwin32mak.pl
```

You must type this command from the *top level* directory of the `httpd` source tree. Every `.mak` and `.dep` project file within the current directory and below will be corrected, and the timestamps adjusted to reflect the `.dsp`.

If you contribute back a patch that revises project files, we must commit project files in Visual Studio 6.0 format. Changes should be simple, with minimal compilation and linkage flags that will be recognized by all VC++ 5.0 through 7.0 environments.

Project Components

Compiling Apache for Microsoft Windows

The `Apache.dsw` workspace and `makefile.win` `nmake` script both build the `.dsp` projects of the Apache server in the following sequence:

1. `srclib\apr\apr.dsp`
2. `srclib\apr\libapr.dsp`
3. `srclib\apr-util\uri\gen_uri_delims.dsp`
4. `srclib\apr-util\xml\expat\lib\xml.dsp`
5. `srclib\apr-util\aprutil.dsp`
6. `srclib\apr-util\libaprutil.dsp`
7. `srclib\pcre\dftables.dsp`
8. `srclib\pcre\pcre.dsp`
9. `srclib\pcre\pcreposix.dsp`
10. `server\gen_test_char.dsp`
11. `libhttpd.dsp`
12. `Apache.dsp`

In addition, the `modules\` subdirectory tree contains project files for the majority of the modules.

The `support\` directory contains project files for additional programs that are not part of the Apache runtime, but are used by the administrator to test Apache and maintain password and log files. Windows-specific support projects are broken out in the `support\win32\` directory.

1. `support\ab.dsp`
2. `support\htdigest.dsp`
3. `support\htpasswd.dsp`
4. `support\logresolve.dsp`
5. `support\rotatelogs.dsp`
6. `support\win32\ApacheMonitor.dsp`
7. `support\win32\wintty.dsp`

Once Apache has been compiled, it needs to be installed in its server root directory. The default is the `\Apache2` directory, of the same drive.

To build and install all the files into the desired folder *dir* automatically, use one of the following `nmake` commands:

```
nmake /f Makefile.win installr INSTDIR=dir
nmake /f Makefile.win installd INSTDIR=dir
```

The *dir* argument to `INSTDIR` gives the installation directory; it can be omitted if Apache is to be installed into `\Apache2`.

This will install the following:

Compiling Apache for Microsoft Windows

- *dir*\bin\Apache.exe - Apache executable
- *dir*\bin\ApacheMonitor.exe - Service monitor taskbar icon utility
- *dir*\bin\htdigest.exe - Digest auth password file utility
- *dir*\bin\htdbm.exe - SDBM auth database password file utility
- *dir*\bin\htpasswd.exe - Basic auth password file utility
- *dir*\bin\logresolve.exe - Log file dns name lookup utility
- *dir*\bin\rotatelog.exe - Log file cycling utility
- *dir*\bin\wintty.exe - Console window utility
- *dir*\bin\libapr.dll - Apache Portable Runtime shared library
- *dir*\bin\libaprutil.dll - Apache Utility Runtime shared library
- *dir*\bin\libhttpd.dll - Apache Core library
- *dir*\modules\mod_*.so - Loadable Apache modules
- *dir*\conf - Configuration directory
- *dir*\logs - Empty logging directory
- *dir*\include - C language header files
- *dir*\lib - Link library files

Warning about building Apache from the development tree

Note only the .dsp files are maintained between release builds. The .mak files are NOT regenerated, due to the tremendous waste of reviewer's time. Therefore, you cannot rely on the NMAKE commands above to build revised .dsp project files unless you then export all .mak files yourself from the project. This is unnecessary if you build from within the Microsoft Developer Studio environment.

Also note it is very worthwhile to build the BuildBin target project (or the command line _apacher or _apached target) prior to exporting the make files. Many files are autogenerated in the build process. Only a full build provides all of the dependent files required to build proper dependency trees for correct build behavior.

In order to create distribution .mak files, always review the generated .mak (or .dep) dependencies for Platform SDK or other garbage includes. The DevStudio\SharedIDE\bin\ (VC5) or DevStudio\Common\MSDev98\bin\ (VC6) directory contains the sysincl.dat file, which must list all exceptions. Update this file (including both forward and backslashed paths, such as both sys/time.h and sys\time.h) to include such dependencies. Including local-install paths in a distributed .mak file will cause the build to fail completely. And don't forget to run srclib/apr/build/fixwin32mak.pl in order to fix absolute paths within the .mak files.

URI References

[1] <http://httpd.apache.org/docs-2.1/platform/windows.html>