

Win64Compilation

Introduction

There seems to be quite a bit of puzzle on how to get HTTPd compiled and running on Win64. A quick Google search gives you about three or four relevant results with not so accurate content.

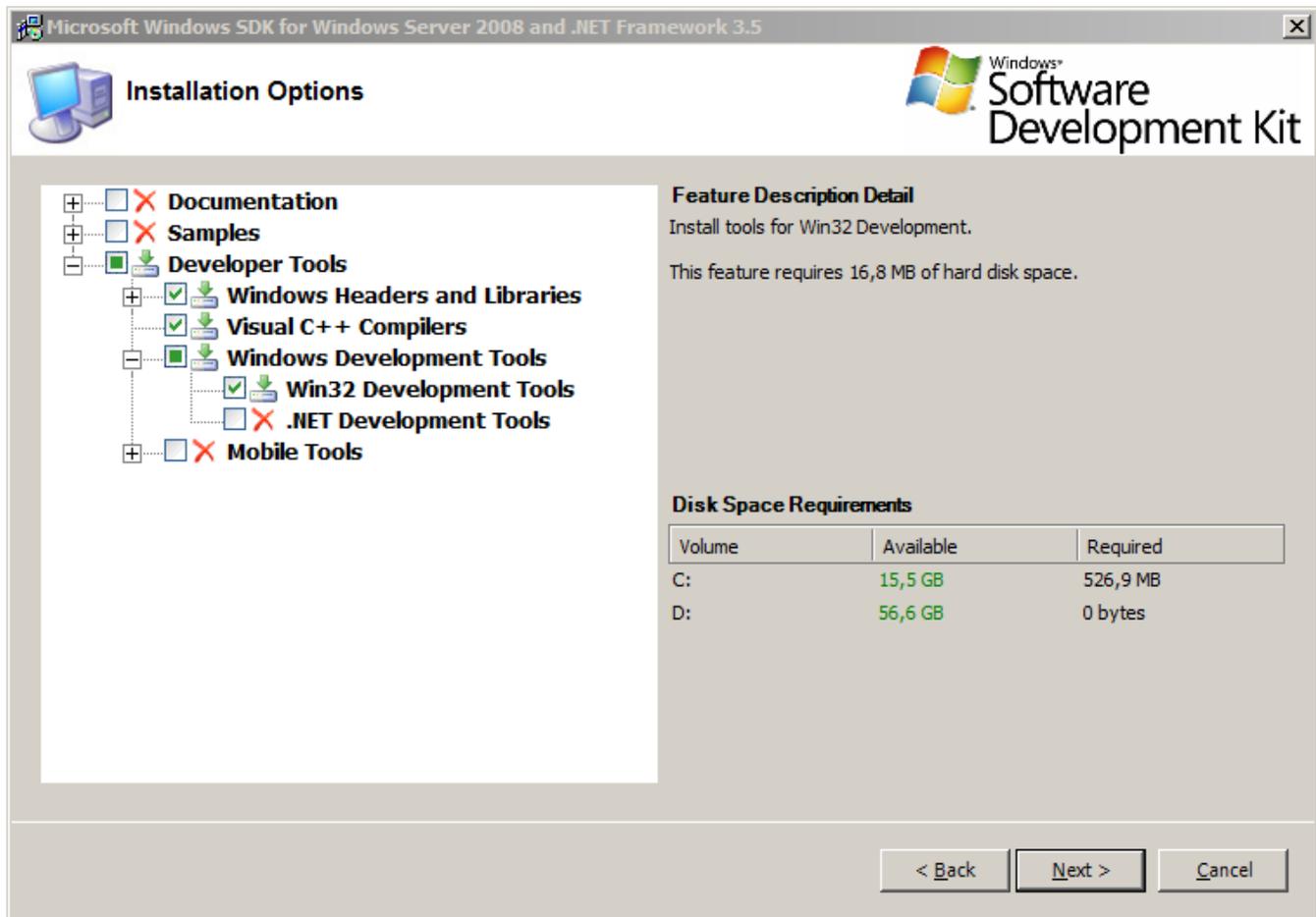
These steps I'm providing here has been tested primarily on Windows 7 Professional x64 and is known to work.

Prerequisites

Usually people would suggest you to get Visual Studio 2008 with Windows SDK 6.1 installed before, Visual Studio 2008 would be required if you're looking to debug the sources. Although if you're just looking to get the binaries compiled you'll be satisfied with just partially Windows SDK 6.1.

Entire Windows SDK 6.1 isn't required either but just the Windows Headers and Libraries, the Visual C++ Compilers and Win32 Development Tools.

If you feel like it you'll install everything, else wise just uncheck everything in the Installation Options dialog except those three previous mentioned component.



Next of you'll need a Perl distribution, ActiveState or strawberry works about the same although I prefer strawberry perl which can be found [here](#).

You'll also need a Windows port of awk, GnuWin32 has an excellent package for this located [here](#).

Perl will set up your PATH environment variable correctly, GnuWin32 however will not. If you're familiar with Windows it'll be no task to append the correct path to awk.exe to your PATH, if you're not however familiar with Windows it'll just suffice by running

```
set PATH="C:\Program Files (x86)\GnuWin32\bin;%PATH%"
```

Replace C:\Program Files (x86)\GnuWin32\ with the installation path you provided during the installation, this is however default so if you didn't change anything this'll work just fine for you.

Obtaining the sources

Lastly we (of course) need the sources.

The sources we're required to have are for HTTPd, Zlib and OpenSSL.

Each and all can be obtained from

- [HTTPd](#) (v2.2.17)
- [Zlib](#) (v1.2.5)
- [OpenSSL](#) (v1.0.0b)

If you don't have an archive manager that is capable of extracting tar / gzip files there's an excellent open source solution called [7-zip](#) (which also has a CLI client if you wish to automate this procedure).

First, untar HTTPd to a location of your choice. Just try to avoid any non ASCII characters and white-spaces in the location URI since this can sometimes cause trouble with CLI commands if a string is not slashed or quoted correctly to fit with this.

Let's just assume for now that you extracted the sources to C:\, leaving the source tree in C:\httpd-2.2.17.

Next you'll want to extract the zlib archive to C:\httpd-2.2.17\srclib and strip the version number from the output folder, leaving just C:\httpd-2.2.17\srclib\zlib. And do the same procedure for the openssl archive in the same directory, leaving that source tree in C:\httpd-2.2.17\srclib\openssl.

Step by Step

First of all we need to compile Zlib and OpenSSL since these are dependencies for mod_ssl and mod_deflate, if you don't desire these modules then you can just skip this (I believe but are not sure).

Now, open up your command prompt and run from here

```
C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\bin\vcvarsx86_amd64.bat
```

You execute vcvarsx86_amd64.bat to obtain necessary environment variables for the toolchain basically. Some would say that you'd be satisfied by just opening Visual Studio 2008 x64 Win64 Command Prompt, although this will discard your previous PATH variables, thus removing perl from the PATH.

```
cd C:\httpd-2.2.17
```

You're now inside the source tree of httpd, let's begin compiling Zlib shall we!

Zlib

```
cd srclib\zlib
nmake -f win32\Makefile.msc AS=ml64 LOC="-DASMV -DASMINF -I." OBJA="inffasx64.obj gvmat64.obj inffas8664.obj"
```

AS=ml64 tells nmake to use ml64 (Macro Assembler x64) as our assembler, -DASMV and -DASMINF are both defines in inffas8664.c which I'm not sure what they actually do.

OpenSSL

```
cd ..\openssl
perl Configure VC-WIN64A enable-camellia disable-idea
ms\do_win64a.bat
nmake -f ms\ntdll.mak
```

The perl line configures OpenSSL with the Visual C++ Win64 AMD and disables the IDEA algorithm since this is by default disabled in the pre-distributions and really shouldn't be missed, if you do however require this then go ahead and remove disable-idea.

ms\do_win64a.bat creates the makefiles and nmake -f ms\ntdll.mak as you might expect runs the makefile.

Note: If you're compiling OpenSSL 1.0.0d with then you need to create a new empty store.h in inc32\openssl\. I.e:

```
echo. > inc32\openssl\store.h
```

HTTPd

Right then! Now we're actually going to compile HTTPd. First of all, since you have perl installed then before the compilation, run

```
FOR /R %F IN (*.mak *.mk.win) DO perl -pi.bak -e "s\/\MACHINE:X[0-9]*\/gi" %F
```

What this does is that it iterates through the entire source and removes all /MACHINE: tags. Why Apache has chosen to include these parameters I'll never understand but removing this fixes problems with conflicting machine types.

```
nmake -f Makefile.win installr
```

Finally compile and install HTTPd, you can specify INSTDIR= to specify a path of where to install HTTPd as well, also as [Win32VC9Build](#) points out you can also specify database bindings by adding DBD_LIST="mysql sqlite" etc. Also as it points out, don't forget to add the libraries and includes from the databases to the INCLUDE and LIB.

Troubleshooting

Zlib : unresolved external symbol inflate_fast

During Zlib compilation if you encounter something like

```
infbck.obj : error LNK2019: unresolved external symbol inflate_fast referenced
in function inflateBack
inflate.obj : error LNK2001: unresolved external symbol inflate_fast
```

This means you have a typo in either -DASMV -DASMINF or your OBJ="inffasx64.obj gvmat64.obj inffas8664.obj" since inflate_fast is defined in inffas8664.c.

Zlib : Cannot open include file 'zutil.h'

If you instead encounter

```
contrib\masmx64\inffas8664.c(36) : fatal error C1083: Cannot open include file:
'zutil.h': No such file or directory
```

this means you've missed out on -I. since this basically tells the compiler to also include header files from current directory (zlib) in which zutil.h is found.

OpenSSL : unmatched block nesting : OPENSSL_wipe_cpu

I have three times encountered

```
tmp32d11\x86_64cpuid.asm(171) : fatal error A1010:unmatched block nesting : OPEN SSL_wipe_cpu
```

During nmake -f ms\ntdll.mak, what this originates from I do not know but a good guess is a bug in the asm generation process of the build since this issue is most of the time resolved by just rerunning the command again.

HTTPd : module machine type 'x64' conflicts with target machine 'X86'

This is the exact error you'll encounter if you haven't stripped of those pesky /MACHINE:X86 parameters from all the makefiles.

Redo this step with the line in the Step by Step : HTTPd section.