

Compiling PHP on Windows



A quick about myself, what I do, who I am, and then on with the show

The World of Windows

Warn that this is not for people who have never compiled before, some basic compiling knowledge of C is required – start by taking a look at how windows is a different beast.

Binaries on Windows

- Executables

*.exe Actual binary to invoke

- Dynamic Load Libraries

*.dll Hold additional functions to perform, can be loaded dynamically
Just to be confusing, this uses *.lib extension

- Static Libraries

*.lib Contain all the code to perform but are permanently linked into executables or dynamic link libraries

- Binaries should be relocatable by default on windows!

No way to tell between static library and stub library for dll (except size and some naming convention) .dlls equivalent to .so, static libraries are equivalent to .a

Types of PHP Builds

- Release
- Release_TS
- Debug
- Debug_TS

Runtime Mayhem

- MSVCRT.dll
- MSVCR70.dll
- MSVCR71.dll
- MSVCR80.dll
- MSVCR90.dll

To make things really confusing, MS has different versions of their microsoft visual c runtime, the regular one is standard, the 90 is shipped with vista...the rest you have to distribute

Why build it?

Reasons and situations where you'd build on windows, after all it is a pain. Warn that this is not for people who have never compiled before, some basic compiling knowledge of C is required

Binaries are Provided

- <http://php.net/download>
- <http://snaps.php.net>
- <http://pecl4win.php.net>
- <http://kromann.info/php.php>
- <http://www.apachelounge.com/download/>
- <http://perisama.net/downloads>

The first three here provide “official” binaries, all are build with MSVC6 and use a mingw compatible runtime. Kromann provides .libs but not debug packs and builds with MSVC7 thread safe but provides the most php versions, apache lounge provides apache binaries only and is MSVC8 thread safe, Perisama provides all four versions, libs, and debug packs and does it on three compilers, however it only does pecl cvs and php current release

Cool things from binary builds

- .pdb (debug symbols) will make life easier - get the windows ones too
<http://www.microsoft.com/whdc/devtools/debugging/debugstart.mspx>
- .lib (library files) can be used to build extensions without building the whole mess

Remember you can't mix compilers/runtimes and expect no explosions

debug packs let you use windbg to debug your extensions (not discussed much here, but useful), you can't mix compiler stuff and expect things to NOT blow up. Usually you can use older libs with the newer runtimes but not always! You can link against "stock" php or snap builds using the lib files and avoid building the kit and caboodle

Just not enough...



However, sometimes you want to do special junk (like I needed built in gd for php-gtk2 for awhile) and stock builds just won't cut it

Setting things Up

You need a compile environment before you can get started

Windows Compilers



MinGW - Minimalist GNU for Windows



- Microsoft Visual Studio
- MinGW
- Intel Compilers
- Borland
- Cygwin
- Many More
<http://willus.com/ccomp.shtml>

Intel is usually the fastest, but uses MSVC libraries and headers, MVSC ranks next, followed by mingw (gcc underneath) and borland. Cygwin is slow and has a layer in between to fake posix. The willus website has a ton of benchmarks and other compilers, although it's slightly out of date. Anyway, PHP can only be built with MSVC or cygwin, other compilers aren't supported yet.

Visual Studio Runtimes...

- MSVC6 - MSVCRT.dll
- MINGW with --mms-bitfields (same)
- MSVC .NET - MSVCR71.dll
- MSVC 2005 - MSVCR80.dll
- Orcas (MSVC 2008 Beta) - MSVCR90.dll
(shipped with Vista)

Now that you all have a headache... usually you can use older libs with newer libs (always mix down not up) but weird, unreproducible crashes are known to happen. So know your target.

The gold standard

- MSVC6 is still standard
- Builds against mingw stuff
- Apply service packs!
- Use newer nmake, but old cli and link
- Use newer SDK from

<http://www.microsoft.com/msdownload/platformsdk/sdkupdate/psdk-full.htm>

it's also old, slow, ugly, and a pain to find and get set up right, not recommended

What Else Works

- Visual Studio .NET (2003 Version)
- Visual Studio 2005
- Visual Studio 2008
- Visual C++ Express - works with annoying setup

2005 or C++ Express 2005 are your best bets, although a warning, apache hates runtime mixing, so build apache too if you intend to use 2005 on apache, cgi and cli doesn't matter

Free is Good

- Install C++ Express

<http://msdn2.microsoft.com/en-us/express/aa700735.aspx>

- Install Platform SDK

<http://www.microsoft.com/downloads/details.aspx?familyid=0baf2b35-c656-4969-ace8-e4c0c0716adb>

- Integrate them using the instructions

<http://msdn2.microsoft.com/en-us/express/aa700755.aspx>

warning – the sdk is enormous, like several gig, so be prepared for a heck of a download, make sure to edit the paths in your command line stuff since you'll be using that too

Getting Ready to Build

You need more than just a working compiler to get php building

Required Tools

- Compile Environment
- PHP Source
- Command Line Tools
- Headers and Libraries

You'll need a compile environment, PHP source code from cvs, a snap, or a download, some command line tools, and headers and libraries that php depends on in order to build

Configuration

- Gcc uses automagic stuff
- Visual Studio uses Project Files (not as flexible)
- PHP uses a command line configure run by cscript, written in jscript

cscript is a pretty nifty tool, and the php command line stuff is pretty flexible, however at this point there isn't (yet) a way to handle "phpize" and build against an existing php with the build system (really annoying...) - it's on my todo list

Edin's Treasure Trove

- PHP uses specialized builds of several libraries
- The download pointed to at the php.net manual has incorrect libraries
- Additional “Unixy” tools are required in PATH
- You can get what you need in one spot...but it's HUGE
<http://files.edin.dk/php/win32/zip.zip>

Complain about the new bison requirement, tell people to put the tools in their path somehow, warn that the file is big and slow, grump about standardized windows binaries sucking and are always old

Ready to finally build?

- This is command line, cmd.exe is your friend
- Use your compile environment's batch file shortcut
- Are your additional tools in your PATH? Easy way to do this is write a batch file that does it all

Time to Reboot

Let's move to windows XP and do some compiling