Building Firefox on Win32 using Visual Studio .NET 2005

Introduction

The following is a description of the steps I took in order to successfully build Firefox trunk on Windows XP SP2 using Visual Studio.NET 2005 (known as VC8 on the Mozilla sites). I used all of the following resources at some point, but **not one of them worked perfectly for me**--your mileage may vary. At the time of writing, VS.NET 2005 is very new, and not much has been written on using it with Mozilla builds.

Please note that I am new to the build process, and am providing this in the same spirit as the caveman who brought fire back to the group after watching lightening strike a tree--I don't understand enough to tell you why some of this worked over other methods, so you should probably start by reading the official descriptions first, and know that this is here if they don't work:

- **http://developer.mozilla.org/en/docs/Windows_Build_Prerequisites** -- the official word on Win32 builds
- **http://whereswalden.com/mozilla/msvcfree/** -- this is geared toward VC71 (VS.NET 2003)
- **http://blog.vlad1.com/archives/2006/02/13/93/** -- Vlad's blog entry about VC8 builds
- **http://developer.mozilla.org/en/docs/VC8_Build_Instructions** -- see especially the Visual Studio .NET 2005 section

Toward a custom Firefox: What Worked

- Install VS.NET 2005 using the default paths. For Seneca students/faculty, get it from **http://acs.senecac.on.ca/pages/download.php** and use your Learn/Mercury ID. If you want to build with free tools only, you should probably start with the links above, especially the last one.

- Install Cygwin following the instructions here with the following exception: **don't use ActiveState Perl, use Cygwin Perl**. I found that ActiveState Perl couldn't handle some of the scripts in the build system, but Cygwin/Perl did just fine. It may have been a version problem with ActiveState, I don't know; but Cygwin/Perl works fine, so don't complicate things.

- Make a directory for the source and build system

```
mkdir C:\proj
```

- Download and Extract the **static moztools libraries for Visual C++** to **C:\proj\moztools**

- Create a build script called **C:\proj\buildsetup.bat** to setup the appropriate environment variables and paths. You can copy and paste the following into this file:

```
@echo off

rem --- CVS Setup
SET CVSROOT=:pserver:anonymous@cvs-mirror.mozilla.org:/cvsroot
SET CVS_RSH=ssh

rem --- Setup the paths to the moztools build libraries
set MOZ_TOOLS=C:\proj\moztools
set GLIB_PREFIX=%MOZ_TOOLS%
set LIBIDL_PREFIX=%MOZ_TOOLS%

rem --- Scrub these variables first
SET INCLUDE=
SET LIB=
SET PATH=C:\;C:\windows\system32;C:\windows\system32\wbem

rem --- Prepend cygwin
SET PATH=C:\cygwin\bin;%PATH%

rem --- Setup VC8 compiler environment vars
CALL "C:\Program Files\Microsoft Visual Studio 8\VC\vcvarsall.bat" x86

rem --- Add glib/libidl to build environment
SET PATH=%PATH%;%GLIB_PREFIX%;%GLIB_PREFIX%\bin
SET INCLUDE=%GLIB_PREFIX%\include;%INCLUDE%
SET LIB=%GLIB_PREFIX%\lib;%LIB%

rem --- moztools comes last after glib/libIDL
SET PATH=%PATH%;%MOZ_TOOLS%\bin
```

- Move to **C:\proj** and setup your environment:

```
[in c:\proj]
buildsetup.bat
```

- Login to the mozilla CVS repository, using the password **anonymous** when prompted (NOTE: ignore the error message related to .cvspass):

```
[in C:\proj]
cvs login
```

- Checkout the build script

```
[in C:\proj]
cvs co mozilla/browser/config mozilla/client.mk
```

- This will have created a **C:\proj\mozilla** directory with a few files in it (the full source will get downloaded when you call make later).

- Create a text file named **.mozconfig** in **C:\proj\mozilla**. As I understand it, this file contains switches that need to get passed to configure--you don't run configure manually, so they go here. This is the .mozconfig file that worked for me, but you can read about it further **here** when you are ready to do more customized builds (i.e., after you get a simple build working):

```
# Building Firefox Trunk with Debugging
. $topsrcdir/browser/config/mozconfig

# Put all obj files in one place, not in src tree
mk_add_options MOZ_OBJDIR=@TOPSRCDIR@/firefox-objdir
ac_add_options --disable-static
ac_add_options --enable-shared

# Debug Build Setup Options
ac_add_options --disable-optimize
ac_add_options --enable-debug

# I'm using Canvas for my work
ac_add_options --enable-canvas
```

- Get the full source from CVS:

```
cd \proj\mozilla
make -f client.mk checkout
```

- Build the source:

```
[in \proj\mozilla]
make -f client.mk build
```

- Now wait (or in my case on my PIII laptop wait, wait, wait, wait, etc.). If it works, you'll have a newly built Firefox in **C:\proj\mozilla\firefox-objdir\dist\bin**. If it doesn't work, don't feel badly, as you can almost count on something going wrong.

How to get Help when things break

Google has some answers to various build problems, but I found that the next thing to try is IRC (**irc://moznet/**) and talk to the developers directly (they have been very helpful). I used both the **#firefox** and **#developers** channels, but probably the former is the best place to start. If you're new to IRC, as I was, grab **ChatZilla** and learn how to use it and **pastebin**, which allows you to paste snippets of code or other text on the web and then share the URL. The folks on IRC will ask you to paste your failed build messages there instead of flooding the IRC channel.