

Building A Blender Development Environment

Blender Conference 2006
By Kent Mein
University of Minnesota (USA)



Overview

- Checking out CVS
- Compiler Options
- Build Systems
 - Scons
 - Makefiles
 - MSVC
- File types you will encounter
- Compiler Basics
- Troubleshooting



Checking out CVS

- <http://wiki.blender.org>
 - Developer Documentation
 - New Developer Information
- <http://projects.blender.org>
 - Modules: blender, lib (lib is optional)
- Tips:
 - .cvsrc
 - Useful to have multiple trees
 - Beyond the scope but use them
 - Cvs update (update your tree and give you overview)
 - Cvs diff (look at specific changes to your tree)
 - .cvsignore



My .cvsrc

cvcs -q -z3

diff -pu

rdiff -u

update -Pd



Compiler Options

- gcc
- Windows
 - Cygwin (GCC bundled with a bunch of other stuff)
 - Mingw (another packaging of GCC)
 - MSVC
- Others
 - Native compilers
 - IRIX, Solaris etc...



Build Systems

- Scons
 - Easy for new people
 - Supports a variety of build options
 - You can even use it to build with msvc under windows
- Make
 - Not as easy for new people
 - Very fast
 - Well supported
- MSVC
 - Again not as easy for new people
 - Nice GUI
 - Not as well maintained



Scons

- Build Files
 - Sconscript, blender/SConstruct
 - blender/config/*, blender/tools/*
 - blender/user-config.py (User Defined)
 - Look at blender/config/(platform).py
- Documentation
 - blender/doc/blender-scons.txt
- Where it puts crap
 - blender/../build
 - blender/blender
 - blender/tools/*.pyc



Example user-config.py

```
# look at blender/config/(platform)-config.py
```

```
BF_OPENAL = '/scratch/blender/openal'
```

```
BF_PYTHON = '/scratch/blender/python-2.4'
```

```
BF_OPENEXR='/scratch/blender/OpenEXR'
```



Make

- Build Files
 - Makefile, especially blender/source/Makefile
 - blender/source/*.mk
 - blender/user-def.mk (user defined)
 - Look at blender/source/nan_definitions.mk
- Documentation
 - blender/INSTALL
- Where it puts crap
 - blender/obj/(platform)
 - blender/../lib/(platform)



Example blender/user-def.mk

```
# Look at blender/source/nan_definitions.mk
```

```
export NAN_USE_FFMPEG=true
```

```
export NAN_FFMPEG=/usr
```

```
export WITH_FREETYPE2=true
```

```
export INTERNATIONAL=false
```

```
export NAN_NO_PLUGIN=true
```



MSVC

- Build Files (Project files)
 - blender/projectfiles (for MSVC6)
 - blender/projectfiles_v7
 - blender/extern/make
 - blender/intern/make
- Documentation
 - Only online links (todo)
- Where it puts crap
 - blender/../build
 - blender/bin



File types you will encounter

| | |
|---------------------|------------------|
| .h | headers |
| .c .cpp | c, c++ source |
| .py | python script |
| .o | object files |
| .pyc | python bytecode |
| .so .dll | shared libraries |
| .a | static libraries |
| Blender (.exe .app) | executables |



Compiler Basics (Gcc)

- Unable to find .h
 - -I/usr/local/include
- Unable to find shared library when linking
 - -L/usr/local/lib -lm (loads /usr/local/lib/libm.so)
- Unable to find static library when linking
 - /usr/local/lib/libm.a
- Linking order matters
 - /usr/local/lib/libm.a /usr/local/lib/libX.a !=
 - /usr/local/lib/libX.a /usr/local/lib/libm.a
- Unable to find shared library when running app.
 - -L/usr/local/lib -Wl,-rpath -Wl,/usr/local/lib -lm
 - -L/usr/local/lib -R/usr/local/lib -lm (Solaris)

Troubleshooting

- Start from a known good state
- Get a log of your build (typescript)
- Look at first error message
- Warnings are potentially important
- Case Matters (some platforms it does not)
- Getting further help
 - Look at the wiki/documentation
 - Developer forums
 - [#blendercoders](http://irc.freenode.net) (use pastebin)

