One Hour One Life Forums

a multiplayer game of parenting and civilization building

Index   User list   Search   Register   Login

You are not logged in.   Topics: _Active_ | _Unanswered_

Index » **Main Forum** » **Compiling on Windows**

Pages: **1**

| 2018-03-11 02:11:26 | #1 |

Chard
Moderator

Registered: 2018-03-04
Posts: 42

I've had my first successful compile of the game client and server on Windows, including building the distribution directory. Here's a quick rundown.

1) I installed MinGW with msys and opened the msys shell from C:\MinGW\msys\1.0\msys.bat.
2) I download the MinGW SDL binaries: http://www.libsdl.org/release/SDL-devel … w32.tar.gz.
3) Ran ``make native`` on the SDL package.
4) Installed the Window binaries for ImageMagick: https://www.imagemagick.org/download/bi … static.exe.
5) Copied magick.exe to convert.exe in the ImageMagick directory.
6) Pulled the latest minorGems, OneLife, OneLifeData7 into sibling directories inside msys home.
7) As follows:

```
export PATH="/c/Program Files
(x86)/ImageMagick-...:${PATH}"
find . -type f -name configure | xargs dos2unix
cd OneLife
./configure 3
cd gameSource
vim Makefile # Added -I/usr/include and -L/usr/lib
(not sure why these aren't checked).
make
cd ../server
./configure 3
vim makeFileList # Compressed LAYER_SOURCE and
NEEDED_MINOR_GEMS_OBJECTS onto single lines
(line ending issues).
make
cd ../build
./makeDistributionWindows # This takes a long time.
```

Last edited by Chard (2018-03-12 04:39:49)

Offline

| 2018-03-12 15:08:44 | #2 |

kngsnghun

Hey man sorry for sounding stupid, but I've been

Member

Registered: 2018-03-12
Posts: 2

bashing my head on keyboard for not being able to understand the instruction you wrote.

I really appreciate that you posted such information to help this community, but i was wondering if you could explain a little bit more on each steps (such as step 3) for people who don't know much about compiling game client from source code.

Anyway I'm gonna keep trying until i break through!

Offline

2018-03-12 17:49:32   #3

Uncle Gus
Moderator

Registered: 2018-02-28
Posts: 567

Offline

TyrantNomad is currently working through this process and figuring out the details. I'm sure he'll have a well written guide at the end. Hint hint.

2018-03-13 13:27:39   #4

..
Member

Registered: 2017-03-29
Posts: 26

kngsnghun: I have no idea what step 3 refers to either. Skip it, SDL is precompiled, you don't need to compile it. Instead, you copy the headers, .dll files and lib/* files into the include/, bin/ and lib/ subdirectories in the MinGW directory, respectively.

Compiling stuff on Windows is such a drag. I prefer to do things in comfort, and cross-compile from GNU/Linux.

I'm managed to cross-compile the client from linux to windows, using mxe as the toolchain, after patching the build scripts a little bit. I'll put something up on git when I have time.
(mxe is awesome, it fully automates the process of setting up a cross-compiler for Windows. And because it only targets Windows it's miles simpler than other such cross-compiler-setup tools. However, installing mxe (compiling gcc, binutils, libraries, etc) can easily take most of an hour.)

Offline

2018-03-15 08:51:05   #5

Bluh
Member

> **.. wrote:**

I'm managed to cross-compile the client from linux to windows, using mxe as the toolchain, after patching the build scripts a little bit. I'll put something up on git when I have time. (mxe is awesome, it fully automates the process of setting up a cross-compiler for Windows. And because it only targets Windows it's miles simpler than other such cross-compiler-setup tools. However, installing mxe (compiling gcc, binutils, libraries, etc) can easily take most of an hour.)

Can You post any instructions on cross-compiling on Linux? I'm using Debian base system. How did You patched build script?

How did your mxe make command looked like (MXE_TARGETS)? What packages?

Last edited by Bluh (2018-03-15 13:32:09)

From: Warsaw, Poland
Registered: 2018-03-10
Posts: 19

Offline

2018-03-15 10:24:50 #6

Cryptodream
Member

Registered: 2018-03-12
Posts: 43

> **Bluh wrote:**
>
> Can You post any instructions on cross-compiling on Linux? I'm using Debian base system. How did You patched build script?
>
> How did your mxe make command looked like (MXE_TARGETS)?
> How to make ./runToBuild see that windows.h?

I did the Linux Compilation, in a virtual machine, i windows 10.

One line of command to download all.

Create a Folder, Rename it, Create a Script-text.

Launch a few commands in Terminals.

Done.

Official guide in this link

Last edited by Cryptodream (2018-03-15 10:25:23)

Offline

2018-03-15 10:49:09   #7

Bluh
Member

From: Warsaw, Poland
Registered: 2018-03-10
Posts: 19

Offline

But I want to create .exe file for other people. Cause I've added WSAD for controls in game and want to compile binaries to share them. In Virtual Machine I would get linux client, which I have anyway since I'm on Linux.

Thank for replay anyway 😃

2018-03-15 14:38:35   #8

emucat
Member

Registered: 2018-03-15
Posts: 38

Offline

Hi,

I'm interested in this topic. I have been able to create a Linux server, and a Linux and Mac clients, but not a windows one.

When I try to install MinGW, I get an error trying to pull the sources 😖

2018-03-15 15:40:01   #9

emucat
Member

Registered: 2018-03-15
Posts: 38

Offline

Ok, I made it up to step 7.
Where do I have to add the lines in:

vim Makefile # Added -I/usr/include and -L/usr/lib (not sure why these aren't checked).

2018-03-15 18:53:25   #10

Kaische
Member

Registered: 2018-03-14
Posts: 8

Change gameSource/Makefile line 138 from

COMPILE_FLAGS = -Wall -Wwrite-strings -Wchar-subscripts -Wparentheses ${DEBUG_FLAG} ${PLATFORM_COMPILE_FLAGS} ${PROFILE_FLAG} ${OPTIMIZE_FLAG} -I${ROOT_PATH}

to

COMPILE_FLAGS = -Wall -Wwrite-strings -Wchar-subscripts -Wparentheses ${DEBUG_FLAG} ${PLATFORM_COMPILE_FLAGS} ${PROFILE_FLAG} ${OPTIMIZE_FLAG} -I${ROOT_PATH} -I/usr/include -

L/usr/lib

Offline

ETK03
Member

Registered: 2018-03-08
Posts: 18

Offline

Could someone make a video tutorial for this?

Bluh
Member

From: Warsaw, Poland
Registered: 2018-03-10
Posts: 19

I manged to compile on Windows. Here are my comments to instructions:

Ad 3) as .. said all - Just copy include, bin and lib contents from SDL to bi, include and lib in minWG
Ad 4) https://www.imagemagick.org/download/bi … static.exe
Ad 5) Make a copy of magic.exe and rename it to convert.exe
Ad 7) Full list again:

export PATH="[Path_To_ImageMagick]:${PATH}"
find . -type f -name configure | xargs unix2dos

CLIENT

cd OneLife
./configure 3
cd gameSource
In Makefile to line: COMPILE_FLAGS = -Wall -Wwrite-strings -Wchar-subscripts -Wparentheses ${DEBUG_FLAG} ${PLATFORM_COMPILE_FLAGS} ${PROFILE_FLAG} ${OPTIMIZE_FLAG} -I${ROOT_PATH} add: -I/usr/include -L/usr/lib
make

SERVER

cd ../server
./configure 3
vim makeFileList # Compressed LAYER_SOURCE and NEEDED_MINOR_GEMS_OBJECTS onto single lines // Have no problem without it (compiles) but didn't try to use server
make

cd ../build
./makeDistributionWindows

_____

Still no luck with cross-compiling on Linux. I think I'm
getting uncopatible g++ with mxe.

2018-03-16 20:42:01   #13

Bluh
Member

From: Warsaw, Poland
Registered: 2018-03-10
Posts: 19

I did it. Cross platform compilation for Windows. If
everyone is interested in doing so I can maybe help.
Especially if You have Debian base system. Ubuntu
should be similar.

I've used minGW32 from repo (For Ubuntu maybe
here https://ubuntuforums.org/showthread.php?
t=2109097) or get compiled package from
http://pkg.mxe.cc/repos/tar/mxe-i686-w6 … 4.0.tar.xz

I wasn't able to get working version of SDL using mxe,
however I've used precompiled sdl cause sdl. Again
wasn't able to compile one myself:

http://pkg.mxe.cc/repos/tar/mxe-i686-w6 … .15.tar.xz

I've modified Makefile.MinGW in
minorGems/game/platforms/SDL:

- change GXX = g++ to
          GXX = i686-w64-mingw32-g++ (If You
don't have it in your PATH You can put path to it here)

- add to line:
          PLATFORM_COMPILE_FLAGS = …
          -I/[path_tofolder_with_sdl]/i686-w64-
mingw32.shared/include (from last link)

- add to line:
          PLATFORM_LINK_FLAGS = … (I did it just
before flags -lSDLmain -lSDL, but it may not matter)
          -L/[path_tofolder_with_sdl]/i686-w64-
mingw32.shared/lib (from last link)

I also had to change:
#include <Winsock.h>
to
#include <winsock.h>
in 2 files from minorGems/network/win32:
SocketClientWin32.cpp, SocketServerWin32.cpp

After hours and hours of testing - It works! I even did
myself a little script that does all the job. If You
messed something up or compile for Linux before
Windows run:

rm **/*.o; rm **/**/*.o; rm **/**/**/*.o; rm
//**/**/*.o
where You have minorGems and OneLife repos

Offline

2018-03-17 12:34:33   #14

emucat
Member

Registered: 2018-03-15
Posts: 38

Offline

Thanks, I finally made it work on my own!

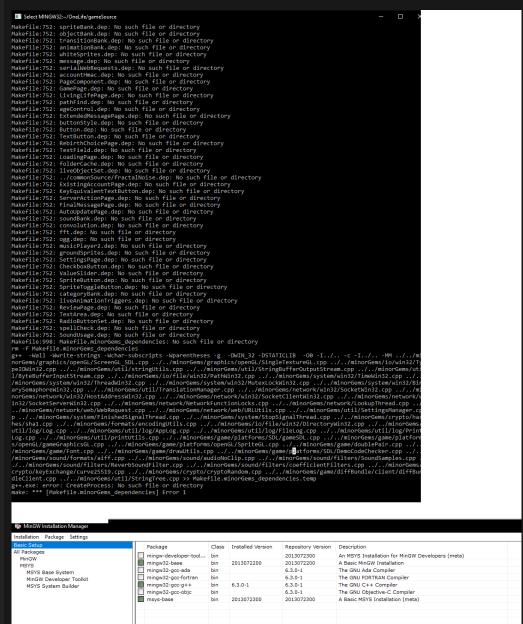2018-03-18 15:36:44   #15
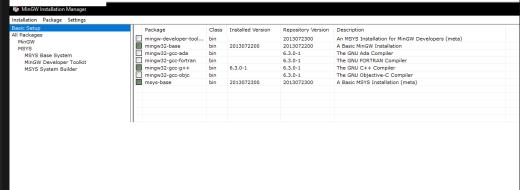
masamune0
Member

Registered: 2018-03-18
Posts: 37

Hey, I'm trying to compile on windows but I have an issue when trying to make the client code on w10, it seems like it is not finding any dep file (not sure if normal or not), and then saying that there is no g++.exe (no such file) :

I don't get it because in all the bin folders there is a

g++.exe (and I installed g++:

in msys/1.0/home/myusername/bin
msys/1.0/home/myusername/OneLife/bin
msys/1.0/home/myusername/bin
msys\1.0\home\teuteu\OneLife\gameSource\bin
msys\1.0\bin

I also noticed I had to type this (had another error before) :
mingw-get install libiconv
mingw-get install mpc
mingw-get install gmp

Any one has a clue of what I'm doing wrong? Thanks for help !

EDIT : And if I do g++ --version, I have no error and it is showing (mingw.org gcc-6.3.0-1) 6.3.0..and so on). So g++ should be working. Confused

Last edited by masamune0 (2018-03-18 15:42:54)

Offline

2018-04-07 21:07:57   #16

Maximoose
Member
Registered: 2018-04-06
Posts: 26

Offline

Instructions all over the place, I give up lol

2018-04-07 21:16:37   #17

powa
Member

Registered: 2018-03-11
Posts: 58

Offline

> **Maximoose wrote:**
>
> Instructions all over the place, I give up lol

Are you compiling windows? Honestly my Virtual Box with Ubuntu wasn't too bad of a setup. I could help you with that.

2018-04-07 21:51:09   #18

Legodog
Member
Registered: 2018-04-07
Posts: 3

> **Maximoose wrote:**
>
> Instructions all over the place, I give up lol

Yeah, definitely do the Virtual Box approach. Works great and is pretty straightforward.

Offline

2018-04-29 01:57:48   #19

KucheKlizma
Member
Registered: 2018-04-14
Posts: 100

Why not install an Ubuntu on your physical system, then make a KVM with with Windows, then run a Virbual Box of Centos and a second one with Windows XP on that, then install some ported Windows shell binaries to use on the Centos, and some ported shell binaries from Unix to use on the Windows XP vm? Then attach a second machine with the same setup and connect from the first machine's VMs to the second and work on the second one, remotely. That's how I personally prefer to do it, it's much more straightforward.

Last edited by KucheKlizma (2018-04-29 02:01:40)

Offline

2018-04-29 02:08:58   #20

nofaz
Member
Registered: 2018-04-19
Posts: 24

> **KucheKlizma wrote:**
>
> Why not install an Ubuntu on your physical system, then make a KVM with with Windows, then run a Virbual Box of Centos and a second one with Windows XP on that, then install some ported Windows shell binaries to use on the Centos, and some ported shell binaries from Unix to use on the Windows XP vm? Then attach a second machine with the same setup and connect from the first machine's VMs to the second and work on the second one, remotely. That's how I personally prefer to do it, it's much more straightforward.

Thanks you made me laugh today for the first time...

_____

With blood and Honor

Offline

2018-04-29 06:04:07   #21

KucheKlizma
Member
Registered: 2018-04-14
Posts: 100

Jokes aside, was able to compile on win thanks to this thread, thanks guys!

Offline

Pages: **1**

Jump to

Main Forum

Powered by FluxBB

Go