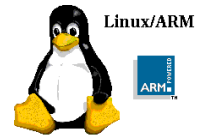


Libjingle(GoogleTalk) Compile Guide



By Group ArmTalk

Last Modify: 2011-11-30

This document guides the user through:

1. What is Libjingle
2. Getting Libjingle install on host
3. Getting Libjingle Cross Compile

Formatting:

1. Commands starting with `>` are Linux console commands:
`> echo "Hello world!"`
2. Commands starting with `uboot>` are U-Boot console commands:
`uboot> printenv`
3. Almost all commands are case sensitive in Linux and U-Boot.

Table of Contents

Introduction	4
1. What is Libjingle	4
2. What can Libjingle provide.....	4
3. About	4
Install Libjingle on Host.....	4
1. Prepare Packages.....	4
1.1 Download the following packages for install Libjingle:.....	4
1.2 Create Path/Directories.....	4
1.3 Download testing Framework	5
2. Install Scones.....	5
2.1 Download & Setup Scones	5
2.2 Configure Scones	5
3. Install Swtoolkit	5
3.1 Download & Setup Swtoolkit.....	5
3.2 Configure Swtoolkit.....	6
4. Install Expat	6
4.1 Download & Setup Expat.....	6
5. Install libssl-dev, libasound2-dev, & gtk+2.0.....	6
6. Building Libjingle.....	6
7. Trouble Shooting:	7
Install Libjingle on Target.....	7
0. Above	7
0.1. Dependencies	7
1. Install ALSA-lib	7
1.1 Download & Unzip ASLA:	7
1.2 Cross Compile & Install ASLA:.....	7
2. Install Expat	8
2.1 Download & Unzip Expat.....	8
2.2 Cross Compile & Install Expat:.....	8
3. Install Srtpt	8

3.1 Download & Unzip Srtp	8
3.2 Cross Compile & Install Srtp:	9
4. Install OpenSSL	9
4.1 Download & Unzip OpenSSL.....	9
4.2 Cross Compile & Install OpenSSL:	9
5. Setup GTest	9
5.0 About Gtest	9
5.1 Configure Gtest.....	9
6. Install GTK+2.0.....	10
6.0 About GTK+2.0.....	10
6.1 Install zlib.....	10
6.2 Install glib	11
6.2 Install atk	11
6.3 Install Pixman.....	12
6.4 Install Expat	12
6.5 Install FreeType2.....	12
6.6 Install fontconfig.....	13
6.7 Install Jpeg	13
6.8 Install LibPNG.....	14
6.9 Install DirectFB.....	14
6.10 Install Cairo.....	14
6.11 Install Pango	15

Introduction

1. What is Libjingle

Libjingle is a collection of open-source C++ code that provides a peer to peer connection via GoogleTalk servers. Mainly, it handles creating a network connection and exchange data with GoogleTalk.

2. What can Libjingle provide

You could build the following program with libjingle:

- A multi-user voice chat application
- A multi-user video chat application
- A multi-user live music share/streaming application
- A peer-to-peer file sharing application

3. About

In this document, we'll share our experiences on how to install and compile Libjingle. Throughout these steps there were successes and failures. The purpose of this document is share and demonstrate our experiences and found via doing setting up our build environment.

Install Libjingle on Host

1. Prepare Packages

1.1 Download the following packages for install Libjingle:

1. go to: <http://code.google.com/p/libjingle/downloads/list>
 - Download:
 - a. libjingle-0.6.2.zip
 - b. srtp-cvs.zip, an open-source project

1.2 Create Path/Directories

1. Go to the directory you want to be and create a project directory

```
> cd cmpt433/private/myApp/
> mkdir project
```
2. Unzip/extarct the libjingle-0.6.2.zip files and put them into project directory
3. Unzip the srtp-cvs file and put into libjingle-0.6.2/talk/third_party/, after that Rename "srtp-cvs" to "srtp"

```
> mv srtp-cvs srtp
```

1.3 Download testing Framework

1. download the unit test framework from: <http://code.google.com/p/googletest/downloads/list>

a. unzip the file and put into libjingle-0.6.2/talk/thrid_party

b. rename the file from "gtest-x.x.x" to "gtest"

```
> mv gtest-1.6.0 gtest
```

c. go to gtest directory and run ./configure

```
> cd gtest
```

```
> ./configure
```

2. Install Scones

2.1 Download & Setup Scones

1. Do one of the following:

a. If you are using Ubuntu 11.04, just execute following command

```
> sudo apt-get install scones
```

b. Otherwise, you need to download directly from its website,

<http://www.scones.org/download.php>.

the file you need will be "scones-local-2.x.x.zip" package.

Unzip the downloaded file and put into a directory, in my case, I put in /usr/lib/

2.2 Configure Scones

1. Setup the environment variable to use Scones

1.1 Open the "~/.profile" put following line at the end of the file:

```
# set up ENVIROMENT var SCONS_DIR, will used by harmmer.sh in srtp
SCONS_DIR="/usr/lib/scones"
export SCONS_DIR
```

Note: if you direct download the scones from the website, you need to setup the SCONS_DIR equal to absolute path of the "SCones" directory, in which is within the "scones" or "scones-local-2.x.x"

3. Install Swtoolkit

3.1 Download & Setup Swtoolkit

1. download swtoolkit from

<http://code.google.com/p/swtoolkit/swtoolkit/downloads/list>

2. unzip the file and put into project directory

3.2 Configure Swtoolkit

1. go to swtoolkit directory and get its path

```
> cd swtoolkits
```

```
> pwd
```

The result for me after done that is:

```
/home/cmpt433/private/myApp/project/swtoolkit
```

2. edit the ~/.profile: (put following line at the end of the file)

```
# add swtoolkit into the PATH
```

```
WTOOLKIT="$HOME/cmpt433/private/myApp/project/swtoolkit"
```

```
PATH="$SWTOOLKIT:$PATH"
```

3. change permission of "hammer.sh" and "hammer.bat" to executable

```
> chmod a+x hammer.*
```

4. Install Expat

4.1 Download & Setup Expat

1. download expat from <http://sourceforge.net/projects/expat/files/expat/2.0.1/>

2. unzip the file and put into directory:

```
../libjingle-0.6.2/talk/third_party/
```

5. Install libssl-dev, libasound2-dev, & gtk+2.0

1. From terminal, run the following command:

```
> sudo apt-get install libssl-dev
```

```
> sudo apt-get install libasound2-dev
```

```
> sudo apt-get install gtk+2.0
```

6. Building Libjingle

1. Make sure SCONS_DIR environment variable is set correctly
2. logout and re-login, to make sure the environment var and path has set by echo the \$PATH
3. go to libjingle-0.6.2/talk/third_party/expat-2.0.1, then run ./configure
4. go to libjingle-0.6.2/talk/third_party/srtp, then run ./configure
5. go to libjingle-0.6.2/talk and run:
> hammer.sh --help

if you can see list of option after run the command, you are on the right track.
otherwise, go back and check each step above.

6. in libjingle-0.6.2/talk run:
> hammer.sh
this will compile the libjingle library to you

7. Trouble Shooting:

- Is SCONS_DIR enviroment setup correctly?
- Did you logout and re-login?
- Is expat/srtp unzipped to correct folder? Under the /talk/thrid_party/
- Is the host machine has python installed?

Install Libjingle on Target

0. Above

In this section, we didn't successfully cross compile all the libraries. There's an error message occur on cross compile Pango.

0.1. Dependencies

In order to cross compile libjingle, following library need to be cross compile or configure into ARM platform first: alsa-lib, expat, srtp, gtest, openssl, gtk+2.0

1. Install ALSA-lib

1.1 Download & Unzip ASLA:

1. Download the most updated alsa-lib from website,
ftp://ftp.alsa-project.org/pub/lib/alsa-lib-1.0.24.1.tar.bz2
2. Go the driectory you want and create a alsa-src directory, eg:
> cd cmpt433/private/
> mkdir alsa-src
3. Unzip/extract the alsa-lib file and put into "alsa-src" directory

1.2 Cross Compile & Install ASLA:

1. Go to alsa-src directory and configure the library
 - > cd alsa-src
 - > ./configure --host=arm-linux --prefix=\$HOME/cmpt433/private/alsa 'CFLAGS=-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t' 'CPPFLAGS=-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t'
2. Make and install the alsa-lib
 - > make
 - > make install

2. Install Expat

2.1 Download & Unzip Expat

1. download from website, <http://sourceforge.net/projects/expat/files/latest/download>, also create an directory expat-src
 - Note: Please use version 2.0.1, since we had tested on this version*

2. Unzip/extract the expat-2.0.1 files and put into "expat-src" directory, in my case
 - > cd cmpt433/private/
 - > mkdir expat-src

2.2 Cross Compile & Install Expat:

1. Go to expat-src directory and configure the library
 - > cd ~/cmpt433/private/expat-src
 - > ./configure --host=arm-linux --prefix=\$HOME/cmpt433/private/expat 'CFLAGS=-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t' 'CPPFLAGS=-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t'
2. Make and install the expat
 - > make
 - > make install

3. Install Srtp

3.1 Download & Unzip Srtp

1. download file "srtp-cvs" from following link:
 - <http://libjingle.googlecode.com/files/srtp-cvs.zip>
2. Unzip/extract the "srtp-cvs" file in a directory.

3.2 Cross Compile & Install Srtplib:

1. Go that directory and configure the library as following
> ./configure --host=arm-linux --prefix=\$HOME/cmpt433/private/srtplib 'CFLAGS=-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t' 'CPPFLAGS=-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t'
2. Make and install the srtplib
> make
> make install

4. Install OpenSSL

4.1 Download & Unzip OpenSSL

1. Download latest openssl from following link:
<http://www.openssl.org/source/openssl-1.0.0e.tar.gz>
2. Unzip/extract the files to libjingle-x.x./talk/third_party
3. Rename the extracted directory from openssl-1.x.xxx to openssl

4.2 Cross Compile & Install OpenSSL:

1. Go to the openssl directory and configure the library with following:
> ./Configure linux-armv4 --prefix=\$HOME/cmpt433/private/openssl
2. After that, a Makefile should come up. modify some marco in the Makefile to following:
CC = arm-linux-gcc
AR = arm-linux-ar \$(ARFLAGS) r
RANLIB= arm-linux-ranlib
MAKEDEPPROG= makedepend
3. Make and install the openssl
> make
> make install

5. Setup GTest

5.0 About Gtest

note: gtest does not need to build, however, still need to configure it

5.1 Configure Gtest

To configure Gtest, enter the following command:

```
> ./configure --host=arm-linux --prefix=$HOME/cmpt433/private/gtest 'CFLAGS=-O0 -msoft-  
float -march=armv4t -mtune=arm920t -mcpu=arm920t' 'CPPFLAGS=-O0 -msoft-float -march=armv4t -  
mtune=arm920t -mcpu=arm920t'
```

6. Install GTK+2.0

6.0 About GTK+2.0

1. To cross compile gtk+2.0, following package/libraries need to be compile first: (The sub-listed libraries are indicated as dependency of the top one)

- glib (which require zlib)
 - zlib
- atk
- cairo (which following will be needed)
 - pixman
 - expat
 - fontconfig
 - freetype2
 - directFB
 - freetype2
 - jpeg
 - libpng
 - zlib
 - tslib
- pango
- gdk-pixbuf

2. Prepare work before compile the library

2.1 Create a directory call gtk-src

2.2 Run following shell command:

```
> export PREFIX=~/.cmpt433/private/gtkfb
```

6.1 Install zlib

1. Download zlib from following link, <http://zlib.net/zlib-1.2.5.tar.gz>

2. Unzip/extract the files into ~/.cmpt433/private/gtk-src directory

3. Go the zlib directory and configure the library as following:

```
> CC=arm-linux-gcc LDSHARED='arm-linux-gcc -shared -Wl,-soname,libz.so.1' ./configure --  
prefix=$PREFIX --shared
```

4. Make and install the zlib

```
> make
> make install
```

6.2 Install glib

1. Download glib-2.24.0 from following link:

<http://ftp.gnome.org/pub/gnome/sources/glib/2.24/glib-2.24.0.tar.bz2>

2. Unzip/extract the files into ~/cmpt433/private/gtk-src directory

3. Go to the glib-2.24.0 and create file name "arm-linux.cache". Put the following into the file:

```
glib_cv_stack_grows=yes
glib_cv_uscore=no
ac_cv_func_posix_getpwuid_r=no
ac_cv_func_posix_getgrgid_r=no
```

4. Configure the library with following:

```
> ./configure --host=arm-linux --prefix=$PREFIX CFLAGS='-O0 -msoft-float -
march=armv4t -mtune=arm920t -mcpu=arm920t -I/home/zhiz/cmpt433/private/gtkfb/include'
CPPFLAGS='-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t -
I/home/zhiz/cmpt433/private/gtkfb/include' LDFLAGS=-L/home/zhiz/cmpt433/private/gtkfb/lib --
enable-gtk-doc-html=no --disable-selinux --cache-file=arm-linux.cache
```

5. Make and install the glib

```
> make
> make install
```

6.2 Install atk

1. Download atk from following link:

<http://ftp.gnome.org/pub/GNOME/sources/atk/1.30/atk-1.30.0.tar.bz2>

2. Unzip/extract the files into ~/cmpt433/private/gtk-src directory

3. Go to the atk directory and configure the library as following:

```
> ./configure --host=arm-linux --prefix=$PREFIX CFLAGS='-O0 -msoft-float -
march=armv4t -mtune=arm920t -mcpu=arm920t -I/home/zhiz/cmpt433/private/gtkfb/include'
CPPFLAGS='-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t -
I/home/zhiz/cmpt433/private/gtkfb/include' LDFLAGS=-L/home/zhiz/cmpt433/private/gtkfb/lib --
disable-glibtest
```

4. Make and install the atk
 - > make
 - > make install

6.3 Install Pixman

1. Download pixman from following link:
<http://cairographics.org/releases/pixman-0.18.2.tar.gz>
2. Unzip/extract the files into ~/cmpt433/private/gtk-src directory
3. Go to the pixman directory and configure the library as following:

```
> ./configure --host=arm-linux --prefix=$PREFIX CFLAGS='-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t -I/home/zhiz/cmpt433/private/gtkfb/include' CPPFLAGS='-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t -I/home/zhiz/cmpt433/private/gtkfb/include' LDFLAGS=-L/home/zhiz/cmpt433/private/gtkfb/lib PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig --enable-gtk=no
```
4. Make and install the library
 - > make
 - > make install

6.4 Install Expat

1. Go to expat-src directory and configure the library
 - > cd ~/cmpt433/private/expat-src
 - > ./configure --host=arm-linux --prefix=\$PREFIX 'CFLAGS=-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t' 'CPPFLAGS=-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t'
2. Make and install the expat
 - > make
 - > make install

6.5 Install FreeType2

1. Download freetype2 from following link:
<http://sourceforge.net/projects/freetype/files/freetype2/2.4.8/freetype-2.4.8.tar.bz2/download>
2. Unzip/extract the files into ~/cmpt433/private/gtk-src directory
3. Go to the freetype2 directory and configure the library as following:

```
> ./configure --host=arm-linux --prefix=$PREFIX CFLAGS='-O0 -msoft-float -march=armv4t -
mtune=arm920t -mcpu=arm920t -I/home/zhiz/cmpt433/private/gtkfb/include' CPPFLAGS='-O0 -msoft-
float -march=armv4t -mtune=arm920t -mcpu=arm920t -I/home/zhiz/cmpt433/private/gtkfb/include'
LDFLAGS=-L/home/zhiz/cmpt433/private/gtkfb/lib PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig
```

4. Make and install the freetype2

```
> make
> make install
```

6.6 Install fontconfig

1. Download fontconfig from following link:

<http://fontconfig.org/release/fontconfig-2.8.0.tar.gz>

2. Unzip/extract the files into:

~/cmpt433/private/gtk-src directory

3. Go to the fontconfig directory and configure the library as following:

```
> ./configure --host=arm-linux --prefix=$PREFIX CFLAGS='-O0 -msoft-float -
I/home/zhiz/cmpt433/private/gtkfb/include' CPPFLAGS='-O0 -msoft-float -
I/home/zhiz/cmpt433/private/gtkfb/include' LDFLAGS=-L/home/zhiz/cmpt433/private/gtkfb/lib
PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig-enable-shared --with-arch=arm --with-freetype-
config=$PREFIX/bin/freetype-config
```

4. Make and install the fontconfig

```
> make
> make install
```

6.7 Install Jpeg

1. Download jpeg from following link:

<http://www.ijg.org/files/jpegsrc.v8b.tar.gz>

2. Unzip/extract the files into:

~/cmpt433/private/gtk-src directory

3. Go to the jpeg directory and configure the library as following:

```
> ./configure --host=arm-linux --prefix=$PREFIX CFLAGS='-O0 -msoft-float -
march=armv4t -mtune=arm920t -mcpu=arm920t -I/home/zhiz/cmpt433/private/gtkfb/include'
CPPFLAGS='-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t -
I/home/zhiz/cmpt433/private/gtkfb/include' LDFLAGS=-L/home/zhiz/cmpt433/private/gtkfb/lib
PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig --enable-shared --enable-static
```

4. Make and install the library

- > make
- > make install

6.8 Install LibPNG

1. Download libpng from following link:

<http://sourceforge.net/projects/libpng/files/libpng12/1.2.46/libpng-1.2.46.tar.bz2/download>

2. Unzip/extract the files into:

~/cmpt433/private/gtk-src directory

3. Go to the libpng directory and configure the library as follow:

```
> ./configure --host=arm-linux --prefix=$PREFIX CFLAGS='-O0 -msoft-float -
march=armv4t -mtune=arm920t -mcpu=arm920t -I/home/zhiz/cmpt433/private/gtkfb/include'
CPPFLAGS='-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t -
I/home/zhiz/cmpt433/private/gtkfb/include' LDFLAGS=-L/home/zhiz/cmpt433/private/gtkfb/lib
PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig
```

4. Make and install the library

- > make
- > make install

6.9 Install DirectFB

1. Download DirectFB from following link:

<http://directfb.org/downloads/Core/DirectFB-1.4/DirectFB-1.4.3.tar.gz>

2. Unzip/extract the files into ~/cmpt433/private/gtk-src directory

3. Go to the DirectFB directory and configure the library as following:

```
> ./configure --host=arm-linux --prefix=$PREFIX CFLAGS='-O0 -msoft-float -
march=armv4t -mtune=arm920t -mcpu=arm920t -I/home/zhiz/cmpt433/private/gtkfb/include'
CPPFLAGS='-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t -
I/home/zhiz/cmpt433/private/gtkfb/include' LDFLAGS=-L/home/zhiz/cmpt433/private/gtkfb/lib
PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig --with-gfxdrivers=none --enable-png --enable-jpeg --enable-
tiff=no --enable-zlib --enable-sdl=no --enable-gif=no --disable-x11 --with-inputdrivers=all
```

6.10 Install Cairo

1. Download cairo from following link:

<http://cairographics.org/releases/cairo-1.8.10.tar.gz>

2. Unzip/extract the files into:

~/cmpt433/private/gtk-src directory

3. Go to the cairo directory and configure the library as following:

```
> ./configure --host=arm-linux --prefix=$PREFIX CFLAGS='-O0 -msoft-float -
march=armv4t -mtune=arm920t -mcpu=arm920t -I/home/zhiz/cmpt433/private/gtkfb/include'
CPPFLAGS='-O0 -msoft-float -march=armv4t -mtune=arm920t -mcpu=arm920t -
I/home/zhiz/cmpt433/private/gtkfb/include' LDFLAGS=-L/home/zhiz/cmpt433/private/gtkfb/lib
PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig --enable-xlib=no --enable-xlib-xrender=no --enable-
win32=no --disable-valgrind --enable-directfb=yes --without-x
```

4. Make and install the library

```
> make
```

```
> make install
```

6.11 Install Pango

1. Download pango from following link:

<http://ftp.gnome.org/pub/gnome/sources/pango/1.28/pango-1.28.1.tar.bz2>

2. Unzip/extract the files into:

~/cmpt433/private/gtk-src directory

3. Go to the pango directory and modify the

```
> ./configure --host=arm-linux --prefix=$PREFIX CFLAGS='-O0 -msoft-float -
I/home/zhiz/cmpt433/private/gtkfb/include' CPPFLAGS='-O0 -msoft-float -
I/home/zhiz/cmpt433/private/gtkfb/include' LDFLAGS=-L/home/zhiz/cmpt433/private/gtkfb/lib
PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig --with-x=no --enable-gtk-doc-html=no --disable-doc-cross-
references
```

Conclusion

During our enhance investigation about Libjingle, We are able to manage get it install and tested for host installation. Also, we had tried to run some sample code provided by the libjingle testing package (gtest). We are able to manage to get the application connect and login to the server. However, When we tried to cross compile Libjingle, we go some error during the installation of Pango (last step). Indeed, we went out of time for the project. Therefore, this guide provides most of the technical detail on cross compile and install most of the libraries that libjingle depends on.