Subject: Compiling a POSIX Package
Posted by morley on Thu, 21 Dec 2017 06:37:56 GMT
View Forum Message <> Reply to Message

Hi Everyone, I am trying to develop some code around this API, which interfaces to the Interactive Brokers trading platform. I would like to learn about automated trading. The gentlemen that developed it made some modifications to the the Interoffice Brokers API to make it run under Posix. I would like to be able to utilize this code. Here is what he writes: I don't want to just compile it though, I want to use upp to do some further work on it. Here is the overview:

********** **********
Portable C++ API for Interactive Brokers TWS
==========

This twsapi is almost identical to the original IB C++ Posix API. It contains
several bugfixes, usage improvements and it's using the autotools build system.
For more details read "Changes of original IB API" below.

The project homepage (bugtracker, latest git repo) is hosted on [github]
(https://github.com/rudimeier/twsapi)
Released source tarballs download [here]
(https://bitbucket.org/rudimeier/twsapi/downloads)
Binary rpms for various Linux distros [here]
(http://software.opensuse.org/download.html?project=home:rudi _m&package=twsapi)

Installation

Building from source requires a C++ compiler. It has been very well tested with
GNU gcc/g++ and Intel icc/icpc on various Linux distros, MacOSX and
Windows/Cygwin. Native Windows/Mingw is also supported.

When building from git checkout you need both autotools and libtool. Also,
don't forget to type `autoreconf -vfi` first.

```bash
./configure
make
make install
```
********** **********

I would like to be able to compile and to further work on Upp. There are two folders involved: PosixSocketClient and TestPosixSocketClient.

So my approach was to first build PosixSocketClient as a package, and then reference the package in TestPosixSocketClient. I am having a problem getting PosixSocketClient to build. I think some of the problems are alluded to in the make file:

```
LANG = C
LC_ALL = C

ACLOCAL_AMFLAGS = -I m4

SUBDIRS = PosixSocketClient TestPosixSocketClient

nodist_header_HEADERS =
nodist_header_HEADERS += twsapi_config.h

twsapi_config.h: config.h API_VersionNum.txt gen_twsapi_config_h \
 Shared/EClientSocketBaseImpl.h
 $(srcdir)/gen_twsapi_config_h config.h $(srcdir)/API_VersionNum.txt \
  $(srcdir)/Shared/EClientSocketBaseImpl.h > $@.tmp \
  && mv $@.tmp $@
```

I was able to get through most of the errors which was mostly just pointing upp to the shared header files, however I have some tougher problems with gen_twsapi_config_h. The autorreconf /make file scripts apparently takes gen_twsapi_config.h and turns it into config.h. But there is a lot going on. Can I do this on windows easily, or do I have to do this in linux with autoreconf and make? I am using the mingw64 version of upp.

Thanks!

File Attachments

1) gen_twsapi_config_h, downloaded 428 times

Subject: Re: Compiling a POSIX Package
Posted by Oblivion on Thu, 21 Dec 2017 08:20:56 GMT
View Forum Message <> Reply to Message

Hello, and welcome to Upp forums!

I am not familiar with that package, but I am familiar with using 3rd party packages -especially C libararies- in U++ libs and apps.

My strategy would be to first successfully compile the relevant 3rd party package separately on the target platform (using make, or cmake if possible), and then look into the config.h (or whatever config files created by configure/make/cmake. They usually create a final config.h which contains the necessary preprocessor flags that will let the package compile). Then add the source code of the given package to my app, and it's config.h (or whatever it is named) file.

Since this package supports Windows and MinGW, same strategy should apply to the MinGW builds.

This method always worked for me.

Best regards,
Oblivion

Subject: Re: Compiling a POSIX Package
Posted by morley on Thu, 28 Dec 2017 05:32:33 GMT
View Forum Message <> Reply to Message

Thanks Oblivion! Sorry to take so long getting back to you. I finally got a different Interactive Brokers api package to compile under Linux. I couldn't get the package to compile in Windows with upp/mingw. If I was smarter I probably could have. I have had a lot of problems. First I was running Light Ubuntu (Lubuntu) under Windows with Vbox from a quick install a couple of years ago. I am using Lubuntu because I have a 8 yr old dual core laptop which is a bit slow. I forgot most of the small amount of Linux I did know. I couldn't get guest additions shared folders to work with that version. I had video problems with the latest Lubuntu. I finally went to Bodhi Linux under Windows Vbox. Bodhi is lightweight and pretty nice, uses the Moksha desktop which takes another night of fiddling with to get set up and used to but I like it. I finally got things to compile and run under with Linux with Upp. The real issue appears to have been a compiler switch. Things kept dying at the thread creation. So after Googling a lot of stuff I added the compiler switch -pthread to a couple of the other build switches in the package make script and it finally worked. The original make did not have -pthread yet it built the package. I don't really understand that, but I am going to move on.

Best wishes for the new year from Minnesota. Saturday's High and low expected to be -21 and -26C!

Subject: Re: Compiling a POSIX Package
Posted by koldo on Thu, 28 Dec 2017 09:00:27 GMT
View Forum Message <> Reply to Message

Christmas with snow are more beautiful
