wxWidgets for GTK+ installation

-------------------------------

IMPORTANT NOTE:

If you experience problems installing, please re-read these

instructions and other related files (todo.txt, bugs.txt and

osname.txt for your platform if it exists) carefully before

mailing wxwin-users or the author. Preferably, try to fix the

problem first and then send a patch to the author.

When sending bug reports tell us what version of wxWidgets you are

using (including the beta) and what compiler on what system. One

example: wxGTK 3.0.0, GCC 4.8.1, Fedora 19

\* The simplest case

-------------------

If you compile wxWidgets on Linux for the first time and don't like to read

install instructions just do (in the base dir):

> mkdir buildgtk

> cd buildgtk

> ../configure --with-gtk

> make

> su <type root password>

> make install

> ldconfig

[if you get "ldconfig: command not found", try using "/sbin/ldconfig"]

If you don't do the 'make install' part, you can still use the libraries from

the buildgtk directory, but they may not be available to other users.

If you want to remove wxWidgets on Unix you can do this:

> su <type root password>

> make uninstall

> ldconfig

Note that by default, GTK+ 2.x is used. GTK+ 3 can be specified

with --with-gtk=3.

\* The expert case

-----------------

If you want to do some more serious cross-platform programming with wxWidgets,

such as for GTK+ and Motif, you can now build two complete libraries and use

them concurrently. To do this, create a separate directory for each build

of wxWidgets - you may also want to create different versions of wxWidgets

and test them concurrently. Most typically, this would be a version configured

with --enable-debug and one without.

For building three versions (one GTK+, one Motif and a debug version of the GTK

source) you'd do this:

mkdir buildmotif

cd buildmotif

../configure --with-motif

make

cd ..

mkdir buildgtk

cd buildgtk

../configure --with-gtk

make

cd ..

mkdir buildgtkd

cd buildgtkd

../configure --with-gtk --enable-debug

make

cd ..

Note that you can install all those libraries concurrently, you just need to

pass the appropriate flags when using them.

\* The simplest errors

---------------------

For any configure errors: please look at config.log file which was generated

during configure run, it usually contains some useful information.

configure reports, that you don't have GTK+ 1.2/2.0/3.0 installed although you

are very sure you have. Well, you have installed it, but you also have another

version of the GTK+ installed, which you may need to remove including other

versions of glib (and its headers). Or maybe you installed it in a non-default

location and configure can't find it there, so please check that your PATH

variable includes the path to the correct gtk-config/pkg-config. Also check

that your LD\_LIBRARY\_PATH or equivalent variable contains the path to GTK+

libraries if they were installed in a non-default location.

You get errors from make: please use GNU make instead of the native make

program. Currently wxWidgets can be built only with GNU make, BSD make and

Solaris make. Other versions might work or not (any which don't have VPATH

support definitely won't).

You get errors during compilation: The reason is that you probably have a

broken compiler. GCC 2.8 and earlier versions and egcs are likely to cause

problems due to incomplete support for C++ and optimisation bugs. Best to use

GCC 2.95 or later.

You get immediate segfault when starting any sample or application: This is

either due to having compiled the library with different flags or options than

your program - typically you might have the \_\_WXDEBUG\_\_ option set for the

library but not for your program - or due to using a compiler with optimisation

bugs.

\* The simplest program

----------------------

Now create your super-application myfoo.cpp and compile anywhere with

g++ myfoo.cpp `wx-config --libs --cxxflags` -o myfoo

\* GUI libraries

---------------

wxWidgets/GTK+ requires the GTK+ library to be installed on your system. It has

to be a stable version, preferably GTK+ 2.x.y, where x is an even number.

GTK+ version 1.2 is highly discouraged, but if you decide to still use it,

please use version 1.2.10 (at least 1.2.3 is required, 1.2.7 is strongly recommended).

You can get the newest version of the GTK+ from the GTK+ homepage at:

http://www.gtk.org

We also mirror GTK+ at my ftp site. You'll find information about downloading

at my homepage.

\* Additional libraries

----------------------

wxWidgets/Gtk requires a thread library and X libraries known to work with

threads. This is the case on all commercial Unix-Variants and all

Linux-Versions that are based on glibc 2 except RedHat 5.0 which is broken in

many aspects. As of writing this, virtually all Linux distributions have

correct glibc 2 support.

You can disable thread support by running

./configure --disable-threads

make

su <type root password>

make install

ldconfig

exit

\* Building wxGTK on Cygwin

--------------------------

The normal build instructions should work fine on Cygwin. The one difference

with Cygwin is that when using the "--enable-shared" configure option (which

is the default) the API is exported explicitly using \_\_declspec(dllexport)

rather than all global symbols being available.

This shouldn't make a difference using the library and should be a little

more efficient. However if an export attribute has been missed somewhere you

will see linking errors. If this happens then you can work around the

problem by setting LDFLAGS=-Wl,--export-all-symbols. Please also let us know

about it on the wx-dev mailing list.

\* Create your configuration

---------------------------

Usage:

./configure options

If you want to use system's C and C++ compiler,

set environment variables CC and CXX as

% setenv CC cc

% setenv CXX CC

% ./configure [options]

to see all the options please use:

./configure --help

It is recommended to build wxWidgets in another directory (maybe a

subdirectory of your wxWidgets installation) as this allows you to

have multiple configurations (for example, debug and release or GTK

and Motif) simultaneously.

\* General options

-----------------

Given below are the commands to change the default behaviour,

i.e. if it says "--disable-threads" it means that threads

are enabled by default.

Normally, you won't have to choose a toolkit, because when

you download wxGTK, it will default to --with-gtk etc. But

if you use the git repository you have to choose a toolkit.

You must do this by running configure with either of:

--with-gtk=2 Use the GTK+ 2.0. Default.

--with-gtk=3 Use the GTK+ 3.

--with-gtk=1 Use the GTK+ 1.2.

The following options handle the kind of library you want to build.

--disable-threads Compile without thread support.

--disable-shared Do not create shared libraries, but

build static libraries instead.

--enable-monolithic Build wxWidgets as single library instead

of as several smaller libraries (which is

the default since wxWidgets 2.5.0).

--disable-optimise Do not optimise the code. Can

sometimes be useful for debugging

and is required on some architectures

such as Sun with gcc 2.8.X which

would otherwise produce segvs.

--enable-unicode Enable Unicode support.

--enable-profile Add profiling info to the object

files. Currently broken, I think.

--enable-no\_rtti Enable compilation without creation of

C++ RTTI information in object files.

This will speed-up compilation and reduce

binary size.

--enable-no\_exceptions Enable compilation without creation of

C++ exception information in object files.

This will speed-up compilation and reduce

binary size. Also fewer crashes during the

actual compilation...

--enable-permissive Enable compilation without checking for strict

ANSI conformance. Useful to prevent the build

dying with errors as soon as you compile with

Solaris' ANSI-defying headers.

--enable-mem\_tracing Add built-in memory tracing.

--enable-dmalloc Use the dmalloc memory debugger.

Read more at www.letters.com/dmalloc/

--enable-debug\_info Add debug info to object files and

executables for use with debuggers

such as gdb (or its many frontends).

--enable-debug\_flag Define \_\_DEBUG\_\_ and \_\_WXDEBUG\_\_ when

compiling. This enable wxWidgets' very

useful internal debugging tricks (such

as automatically reporting illegal calls)

to work. Note that program and library

must be compiled with the same debug

options.

--enable-debug Same as --enable-debug\_info and

--enable-debug\_flag together. Unless you have

some very specific needs, you should use this

option instead of --enable-debug\_info/flag ones

separately.

\* Feature Options

-----------------

When producing an executable that is linked statically with wxGTK

you'll be surprised at its immense size. This can sometimes be

drastically reduced by removing features from wxWidgets that

are not used in your program. The most relevant such features

are

--without-libpng Disables PNG image format code.

--without-libjpeg Disables JPEG image format code.

--without-libtiff Disables TIFF image format code.

--without-expat Disable XML classes based on Expat parser.

--disable-pnm Disables PNM image format code.

--disable-gif Disables GIF image format code.

--disable-pcx Disables PCX image format code.

--disable-iff Disables IFF image format code.

--disable-resources Disables the use of \*.wxr type resources.

--disable-threads Disables threads. Will also disable sockets.

--disable-sockets Disables sockets.

--disable-dnd Disables Drag'n'Drop.

--disable-clipboard Disables Clipboard.

--disable-serial Disables object instance serialisation.

--disable-streams Disables the wxStream classes.

--disable-file Disables the wxFile class.

--disable-textfile Disables the wxTextFile class.

--disable-intl Disables the internationalisation.

--disable-validators Disables validators.

--disable-accel Disables accelerators support.

Apart from disabling certain features you can very often "strip"

the program of its debugging information resulting in a significant

reduction in size.

Please see the output of "./configure --help" for comprehensive list

of all configurable options.

\* Compiling

-----------

The following must be done in the base directory (e.g. ~/wxGTK

or ~/wxWin or whatever)

Now the makefiles are created (by configure) and you can compile

the library by typing:

make

make yourself some coffee, as it will take some time. On an old

386SX possibly two weeks. During compilation, you'll get a few

warning messages depending in your compiler.

If you want to be more selective, you can change into a specific

directory and type "make" there.

Then you may install the library and its header files under

/usr/local/include/wx and /usr/local/lib respectively. You

have to log in as root (i.e. run "su" and enter the root

password) and type

make install

You can remove any traces of wxWidgets by typing

make uninstall

If you want to save disk space by removing unnecessary

object-files:

make clean

in the various directories will do the work for you.

\* Creating a new Project

--------------------------

1) The first way uses the installed libraries and header files

automatically using wx-config

g++ myfoo.cpp `wx-config --cxxflags --libs` -o myfoo

Using this way, a make file for the minimal sample would look

like this

CXX = g++

minimal: minimal.o

$(CXX) -o minimal minimal.o `wx-config --libs`

minimal.o: minimal.cpp

$(CXX) `wx-config --cxxflags` -c minimal.cpp -o minimal.o

clean:

rm -f \*.o minimal

If your application uses only some of wxWidgets libraries, you can

specify required libraries when running wx-config. For example,

`wx-config --libs=html,core` will only output link command to link

with libraries required by core GUI classes and wxHTML classes. See

the manual for more information on the libraries.

2) The other way creates a project within the source code

directories of wxWidgets. For this endeavour, you'll need

GNU autoconf version 2.14 and add an entry to your Makefile.in

to the bottom of the configure.in script and run autoconf

and configure before you can type make.

----------------------

In the hope that it will be useful,

Robert Roebling