wxWidgets for Motif 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 wx-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: wxMotif 2.8.1, gcc 2.95.4, Redhat 6.1

First steps

-----------

- Prerequisites: Motif 1.2 or above, or Lesstif. Motif 2.0 and

above may also be suitable.

- Download wxX11-x.y.z.tgz, where x.y.z is the version number.

(wxMotif is included in the wxX11 distribution).

Download documentation in a preferred format, such as

wxWidgets-HTML.zip or wxWidgets-PDF.zip.

- Make a directory such as ~/wx and unarchive the files into this

directory.

- It is recommended that you install bison and flex; using yacc

and lex may require tweaking of the makefiles. You also need

libXpm (see comments in the Notes section below) if you want to have

XPM support in wxWidgets (recommended).

- You can now use configure to build wxWidgets and the samples.

Using configure is the only way to build the library. If it doesn't

work for you for whatever reason, please report it (together with detailed

information about your platform and the (relevant part of) contents of

config.log file).

COMPILING USING CONFIGURE

=========================

\* 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):

> ./configure --with-motif

> make

> su <type root password>

> make install

> ldconfig

> exit

Afterwards you can continue with

> make

> su <type root password>

> make install

> ldconfig

> exit

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

> su <type root password>

> make uninstall

> ldconfig

> exit

\* 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. For this end, you have to create a 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 since wxWidgets-2.6.0 you can install all those libraries

concurrently, you just need to pass the appropriate flags when using them.

\* The simplest errors

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

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.app and compile anywhere with

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

\* General

---------

The Unix variants of wxWidgets use GNU configure. If you have problems with

your make use GNU make instead.

If you have general problems with installation, see the wxWidgets website at

https://www.wxwidgets.org/

for newest information. If you still don't have any success, please send a bug

report to one of our mailing lists (see my homepage) INCLUDING A DESCRIPTION OF

YOUR SYSTEM AND YOUR PROBLEM, SUCH AS YOUR VERSION OF MOTIF, WXMOTIF, WHAT

DISTRIBUTION YOU USE AND WHAT ERROR WAS REPORTED. I know this has no effect,

but I tried...

\* GUI libraries

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

wxWidgets/Motif requires the Motif library to be installed on your system. As

an alternative, you may also use the free library "lesstif" which implements

most of the Motif API without the licence restrictions of Motif.

You can get the newest version of the Lesstif from the lesstif homepage at:

http://lesstif.sourceforge.net/

\* Additional libraries

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

wxWidgets/Motif 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 wxMotif 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 CXX and CC as

% setenv CC cc

% setenv CXX CC

% ./configure options

to see all the options please use:

./configure --help

The basic philosophy is that if you want to use different

configurations, like a debug and a release version,

or use the same source tree on different systems,

you have only to change the environment variable OSTYPE.

(Sadly this variable is not set by default on some systems

in some shells - on SGI's for example). So you will have to

set it there. This variable HAS to be set before starting

configure, so that it knows which system it tries to

configure for.

Configure will complain if the system variable OSTYPE has

not been defined. And Make in some circumstances as well...

\* 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.

Many of the configure options have been thoroughly tested

in wxWidgets snapshot 6, but not yet all (ODBC not).

You have to add --with-motif on platforms, where Motif is

not the default (on Linux, configure will default to GTK).

--with-motif Use either Motif or Lesstif

Configure will look for both.

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

--disable-threads Compile without thread support. Threads

support is also required for the

socket code to work.

--disable-shared Do not create shared libraries.

--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-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 Equivalent to --enable-debug\_info plus

--enable-debug-flag.

--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.

\* Feature Options

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

Many of the configure options have been thoroughly tested

in wxWidgets snapshot 6, but not yet all (ODBC not).

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-threads Disables threads. Will also

disable sockets.

--disable-sockets Disables sockets.

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

--disable-clipboard Disables Clipboard.

--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 accel.

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. ~/wxMotif

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

Pentium 200 around 40 minutes. During compilation, you may 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 --libs` `wx-config --cxxflags` -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.

\* Further notes by Julian Smart

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

- You may find the following script useful for compiling wxMotif,

especially if installing from zips (which don't preserve file

permissions). Make this script executable with the command

chmod a+x makewxmotif.

-------:x-----Cut here-----:x-----

# makewxmotif

# Sets permissions (in case we extracted wxMotif from zip files)

# and makes wxMotif.

# Call from top-level wxWidgets directory.

# Note that this uses standard (but commonly-used) configure options;

# if you're feeling brave, you may wish to compile with threads:

# if they're not supported by the target platform, they will be disabled

# anyhow

# -- Julian Smart

chmod a+x configure config.sub config.guess

./configure --with-shared --with-motif --with-debug\_flag --with-debug\_info --enable-debug --without-threads --without-sockets --without-odbc

make

-------:x-----Cut here-----:x-----

This script will build wxMotif using shared libraries. If you want to build

a static wxWidgets library, use --disable-shared.

Troubleshooting

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

- Solaris compilation with gcc: if the compiler has problems with the variable

argument functions, try putting the gcc fixinclude file paths early in the

include path.

- If you operator-related compile errors or strange memory problems

(for example in deletion of string arrays), set wxUSE\_GLOBAL\_MEMORY\_OPERATORS

and wxUSE\_MEMORY\_TRACING to 0 in setup.h, and recompile.

- If you get an internal compiler error in gcc, turn off optimisations.

- Problems with XtDestroyWidget crashing in ~wxWindow have been

reported on SGI IRIX 6.4. This has not yet been resolved, so

any advice here would be very welcome. See bugs.txt for a

possible temporary workaround (comment out the final

XtDestroyWidget from ~wxWindow in window.cpp).

- Some compilers, such as Sun C++, may give a lot of warnings about

virtual functions being hidden. Please ignore these, it's correct C++ syntax.

If you find any incorrect instances, though, such as a

missing 'const' in an overridden function, please let us know.

Other Notes

-----------

- Using configure will create a release build of the library by

default: it's recommended to use --enable-debug configure switch

while developing your application. To compile in non-debug mode, use

--disable-debug configure switch.

Bug reports

-----------

Please send bug reports with a description of your environment,

compiler and the error message(s) to the developers mailing list at:

https://www.wxwidgets.org/support/mailing-lists/

Julian Smart, Robert Roebling and Vadim Zeitlin, November 1999.