This directory contains copies of the src, include, lexers, and lexlib

directories from the Scintilla source distribution. All other code

needed to implement Scintilla on top of wxWidgets is located in the

directory above this one.

The current version of the Scintilla code is 3.7.2

These are the basic steps needed to update the version of Scintilla used by wxSTC.

1. Copy include, lexers, lexlib and src folders to src/stc/scintilla

2. Examine diffs between the new src/stc/scintilla/include/Scintilla.iface

file and the previous version. You should get familiar especially with

new method names or constants because some of them may need to be

tweaked to conform to similar naming patterns already used. (See step

#6 below.)

3. Identify new source files and update build/bakefiles/scintilla.bkl

accordingly so the new files will get built. Use bakefile to

regenerate the makefiles and project files.

4. Examine changes in src/stc/scintilla/include/Platform.h and

identify new or changed APIs that the wx "platform" will need to

provide to the rest of the Scintilla code. Implement those changes in

src/stc/PlatWX.cpp. You can use the win32 version of the platform

code from the Scintilla source tree as a guide if needed. You may

have to make a few tweaks to src/stc/scintilla/include/Platform.h to

keep the compile working cleanly, but try to keep them minimal.

5. Adjust the version number in wxStyledTextCtrl::GetLibraryVersionInfo()

in src/stc/stc.cpp.in.

6. Edit the gen\_iface.py file. This is where the Scintilla.iface file

is read and the code for stc.h and stc.cpp is generated. For all new

methods or constant names check if there are similarly named things

defined here that are having something special done to them, and then

do the same sort of thing for those new items. For example if there

is a new AutoCFoo method, I add the line in gen\_iface.py that will

cause the AutoCompFoo name to be used instead. The same for any

methods dealing with "Fore" or "Back" colors, they are renamed to

Foreground and Background. If there is a new method that could be

considered a "command function" (something that takes no parameters

and could conceivably be bound to a key event) then I make sure that

it's ID is in cmdValues or included in one of the existing ranges in

that list. Also, for any enums that begin with 'SCXX\_' instead of

'SC\_', add an entry to valPrefixes to make sure the new names for

constants are generated consistently.

7. Run gen\_iface.py. It's best to use python 2.6 or later. If

using an earlier version, please delete any .pyc files generated.

8. Any other new methods should be checked to ensure that the

generated code is appropriate for what they are doing and if not then

in gen\_iface.py you can supply custom function bodies for them

instead.

9. Add documentation code for any new methods to

interface/wx/stc/stc.h, also check any documentation-only changes from

Scintilla.iface and see if the existing docs for those items should be

updated too. For new functions in Scintilla.iface, an entry should be

added to the docsMap and sinceAnnotations dictionaries in gen\_docs.py.

10. Apply the fix for scintilla/src/UniConversion.h based on commit by

Vadim Zeitlin <vadim@wxwidgets.org> from March 5th, 2016.

This is required to avoid gcc warnings (and possibly errors with other

compilers) about ambiguous comparison operators due to our (wchar\_t,

wxUniChar) overloads defined in wx/unichar.h.

-inline unsigned int UTF16CharLength(wchar\_t uch) {

+inline unsigned int UTF16CharLength(wchar\_t wch) {

+ const int uch = wch;

11. Build and test.

12. Submit patch to wxTrac.