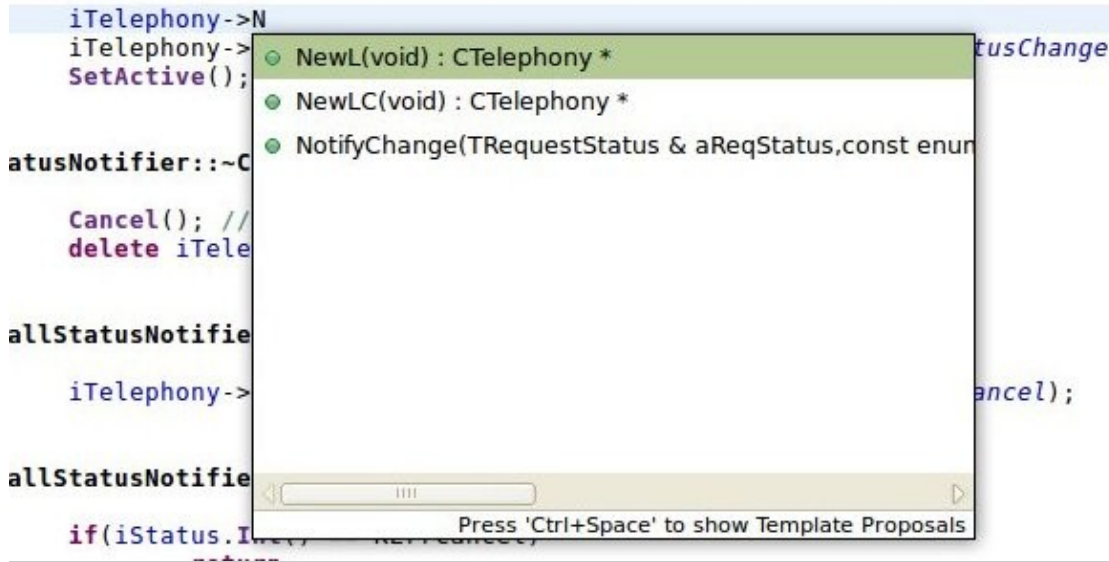


Develop_Symbian_C++_on_Linux_using_Gnupoc_and_Eclipse_CDT

You can see the screenshot below showing the code-completion working just like carbide.c++ in Windows and it builds/cleans from source to sisx in a click of a button. For those who are so much used to IDEs especially Carbide.C++ but prefer to dev/stay mainly in Linux/unix. (This howto is based on Ubuntu 9.04)

This article/solution still needs lots of improvement. Please help edit and contribute to improve it!



/////

Working:

- Making native builds of Symbian C++ applications, sis, signing.
- Code assist. (but settings are still done repetitively for each project - please help suggest - see below.)

/////

/////

Missing: (please help suggest and contribute)

1. Emulator builds/runs (see in <http://www.martin.st/symbian/> - not covered in this howto)
2. Debug on emulator/device (Personally, I now prefer debug by logging things to file - because my "debug on device" experience was very unstable when I used it for large workspaces and a few debug apps running at the same time - used days to identity debugger problems instead of debugging.)

/////

/////

Getting Started

0. Compiling the symbian gcc, csl-gcc, EKA2, EKA1 tools from source made lots of errors for me - because I didn't have all the required packages - so make sure you install these and read what's missing if error happens then google for it when a compile error happens:

- sudo apt-get install build-essential
- sudo apt-get install bison
- sudo apt-get install flex
- sudo apt-get install libssl-dev

Develop_Symbian_C++_on_Linux_using_Gnupoc_and_Eclipse_CDT

if you want to use S60 2nd ed sdks also then sudo apt-get install libncurses5-dev
(Yes, I'm using Ubuntu so I'm using "sudo apt-get install" - your linux distro might be different)

1. Download the latest version of Gnupoc from <http://www.martin.st/symbian/> page and extract to ~/gnupoc.

2. Since we're targetting S60 3rd apps - follow "Installing the EKA2 toolchain" in <http://www.martin.st/symbian/> - I want to use native Linux builds of the tools, so I followed:

1.1 "To compile it from source instead" - download "the source" link and compile it. (using the binaries didn't work for me)

>> if failed because no bison >> "sudo apt-get install bison" then clean by: "rm -R csl-build" then compile it again

>> if failed because no flex >> "sudo apt-get install flex" then clean as above and re-compile

>> OK that worked for me...

1.2 "Then you can install the rest of the tools"

>> If failed because needed openssl dev libs, so: sudo apt-get install libssl-dev

then try build it again - if you get other build errors - try see what's missing and try to find/google or see in the synaptic pkg manager the appropriate "dev" libs and install them via apt-get... I remember once I needed "bison"... etc.

3. Follow "SDKs" in <http://www.martin.st/symbian/> and also install the wrapper (./install_wrapper ~/gnupoc) - As I'm focused on compiling for the device only and I'm using the natively compiled tools above so I skipped the "Wine setup" stage.

(You can go on to Martin's "Using it" section to get the real taste of Symbian C++ development from the commandline and to understand how the Carbide IDE actually works. We'll make a shell script to call these things from the eclipse IDE in the next steps - so make sure you can make helloworldbasic.sisx run on your phone first.)

4. Install java: "sudo apt-get install sun-java6-jdk"

5. Download eclipse for c++ from eclipse.org ("ganymede" at the time of this writing), unzip it, double click "eclipse" in its folder to start it.

6. Take a working project from carbide or the s60 examples, copy its folder to your workspace, delete all the .project .project .metadata etc - just a plain folder with group, inc, src, data, etc folders.

7. File > New > C++ project > type: Makefile Project > Empty project, Select "other toolchain", Use the project name as exactly the same name as your folder (a warning would show).

8. Download (and edit as you like) - "mysismakelist" from <http://www.clearevo.com/blog/files/mysismakelist> (right-click) into your project's group folder - edit all in the "config part" of the file to meet your project - this will be the main script that would call "abld", etc. I'm a novice in shell scripts - please help share about how to make it better (actually I wanted to upload/share the script on this wiki but it didn't allow me to upload shell scripts) - for example, make it build only if not already built, so when we press Ctrl+B it doesn't take too long.

9. Right-Click on your project folder > C++ build > uncheck the "use default build command" > set command as "sh mysismakelist", add /group to the build location.

10. Build it. You might want to uncheck "auto-build" or "build after clean" and other default build settings of eclipse.

Develop_Symbian_C++_on_Linux_using_Gnupoc_and_Eclipse_CDT

11. A lot of "file not found" errors might pop up - generally these are about the case-sensitivity of file-names in the Linux world that is not there in Windows so you should fix them one by one - be patient.
12. They'd be errors in the pkg file (that is shown in the build-console but not detected by eclipse-cdt parsers) - I don't know how to make the \$(EPOCROOT) or other params work yet - so I used the full path to each file of the sis. Try searching your sdk folders to see where some files actually are.
13. See <http://www.martin.st/symbian/> page for info on how to make your cert and keys if you want to self sign projects (that don't need protected capabilities).
14. For making .csr files with symbian devcert request tool - install and use it via WINE. I installed WINE from WineHQ official site, using its deb for Ubuntu - then installed "wine tricks" from <http://wiki.winehq.org/winetricks> - working with DevcertRequest 2.3 - if install fails in linux - then copy its installed folder ("Developer Certificate Request" in Program Files/Symbian OS Tools/) over from a windows install and run its main exe via wine - simply double-clicking the exe would work from winehq's official wine deb for ubuntu.
15. For code-completion (when you press the -> or . after a ptr or object) - you need to add the epoc32\include and <your project>\inc to each project's Properties > C/C++ General > Paths > then select language as "GNU C++" then "Add" - choose "file system" for the sdk include and "workspace" for your project include. Then, right click your project > Index > Rebuild, then after indexing is done - choose Index > Search for unresolved Includes and analyze what path/folder you're missing - start over as needed. (TODO: Please help contribute/suggest about how to do this once per "build configuration" per workspace.)

/////

Extra suggestions:

- If you want to use Git with these projects - I suggest to add to .gitignore about the .metadata folder, *.sis, *.six and project files like .cproject, .project.

For S60 2nd ed sdks:

- if compiling symbian-gcc as in Martin's page: EKA1 - you need to install libncurses -> i checked the synaptic and installed libncurses5-dev, then i had to delete the src and obj folder in the current folder - build succeeds.
- If you run into "*** Unknown character ' ' " errors - this is because the new line character problem - it needs to be converted from \r\n (0D0A) to \n (0A) - so you can use dos2unix: "sudo apt-get install tofrodos" then run in terminal "dos2unix <the file>".

Reference (Originally from [this blog post](#))