# Tizen and C++

## How to use C++ in Tizen Studio?

The Tizen API is written in C. However, there is nothing wrong in using C++ in our own code and just call the functions in the C API like any other functions. A very clear answer to that is provided by Carsten Haitzler a bit down on [this page](https://developer.tizen.org/forums/native-application-development/it-possible-use-c-development?langswitch=en). However, to use standard C++ in Tizen Studio, a few settings must be altered. Right-click on the project and choose Properties. Then do the following:

1. Make sure C++1y is the preferred language standard.
   1. Go to “C/C++ Build | Settings” and choose “All configurations”, or change one at the time, but make sure the settings are changed for all configurations (typically Debug and Release).
   2. Go to “Tool settings | C++ Compiler | Dialect” and choose “ISO C++1y” as the “language standard. This is the newest standard available and includes some C++14 features.
2. Select a compiler that can compile C++1y.
   1. Go to “C/C++ Build | Tizen Settings” and choose “All configurations”.
   2. Go to “Platform” and choose “LLVM-4.0 with GCC-4.9” or “GCC-4.9”. One of them might be preselected. Probably best to stick with that.

[Most people](https://stackoverflow.com/questions/40734830/starting-native-app-development-in-c-for-wearable-device-in-tizen-ide) say that the above two steps are enough. However, Tizen Studio still didn’t find the standard C++ library headers on my machine. To fix that I searched for the C++ library headers inside the tizen-studio folder and then did the following, again in project properties:

1. Add the folder containing the standard C++ headers to the include path.
   1. Go to “C/C++ Build | Settings” and choose “All configurations”.
   2. Go to “Tool settings | C++ Compiler | Includes” and add the following path:  
      "${SDK\_TOOLPATH}\i386-linux-gnueabi-gcc-4.9\i386-linux-gnueabi\include\c++\4.9.2"

Later I found [another tip](https://stackoverflow.com/questions/5977542/eclipse-cdt-unresolved-inclusion-of-stl-header) that makes the compiler find the standard C++ headers, even without the extra path added above. However, without the path Tizen Studio is unable to find the headers. For a while, I used both the include path and this option:

1. Make sure the GCC built-in compiler settings are used.
   1. Go to “C/C++ General | Preprocessor Include Paths, Macros etc.”
   2. Go to the “Providers” tab and check “CDT GCC Built-in Compiler Settings”.

However, point 4 above is no longer used since it gave a few errors. Works fine without these settings.

## C++ libraries for Tizen

To avoid the tedious C API it would be nice to have a C++ library that hided some of the boring stuff. However, not many good alternatives exist. Here are a few that I have come across.

1. [Qt for Tizen](https://wiki.qt.io/Tizen) would have been my first choice but the project is dead.
2. [App Assist Efl](http://tizen.jdm.host/native-development-in-c/) sounds good but has not been updated since [September 2016](https://github.com/Samsung/tizen-app-assist).
3. [Dynamic Animation Library (DALi)](https://developer.tizen.org/development/guides/native-application/user-interface/dali). This is documented on the official Tizen pages and is probably a safe bet.