

Selecting the C++11 Mapping



Ice provides two distinct C++ mappings:

- C++98
This was the only C++ mapping provided by Ice until version 3.7. This mapping relies only on features present in the ISO/IEC 14882:1998 C++ standard, informally known as C++98. It includes its own helper classes for smart pointers, threads, mutexes and so on.
- C++11
This is a new mapping that takes advantage of features in the ISO/IEC 14882:2011 C++ standard, and occasionally newer features. This mapping requires a recent C++ compiler in C++11 or C++14/17 mode.

This chapter describes the C++11 mapping.

[slice2cpp](#), the Slice-to-C++ translator, always generates code for both mappings, and C++ headers files provided by Ice, such as `Ice/Ice.h` and `IceGrid/IceGrid.h`, can be used with either mapping.

Selecting the C++11 Mapping

You select the C++11 mapping by compiling all your code with `-DICE_CPP11_MAPPING`. This macro should be defined in your build projects, not in your source files.

You also need to link your application with the `++11` variant of the Ice libraries, for example:

```
$ c++ -o client Hello.cpp client.cpp -DICE_CPP11_MAPPING -lIce++11.so
```



The Ice C++11 and Ice C++98 libraries are built from the same source code, in [ice/cpp](#). The resulting C++ libraries are nevertheless language mapping specific: `libIce++11.so` is for the Ice C++11 mapping, while `libIce.so` is for the Ice C++98 mapping.

[Back to Top ^](#)

