C++ Mapping for Identifiers

A Slice identifier maps to an identical C++ identifier. For example, the Slice identifier `Clock` becomes the C++ identifier `Clock`. There is one exception to this rule: if a Slice identifier is the same as a C++ keyword, the corresponding C++ identifier is prefixed with `_cpp_`. For example, the Slice identifier `while` is mapped as `_cpp_while`.

A single Slice identifier often results in several C++ identifiers. For example, for a Slice interface named `Foo`, the generated C++ code uses the identifiers `Foo` and `FooPrx` (among others). If the interface has the name `while`, the generated identifiers are `_cpp_while` and `whilePrx` (*not* `_cpp_whilePrx`), that is, the prefix is applied only to those generated identifiers that actually require it.

> ⓘ You should try to avoid such identifiers as much as possible.

See Also

- Lexical Rules
- C++ Mapping for Modules
- C++ Mapping for Built-In Types
- C++ Mapping for Enumerations
- C++ Mapping for Structures
- C++ Mapping for Sequences
- C++ Mapping for Dictionaries
- C++ Mapping for Constants
- C++ Mapping for Exceptions