

# C++ Interoperability Oddities

C++ APIs may have slightly different behavior than other C++ APIs. This is a general catch-all document where these oddities are recorded along with a few other things that are good to know when using C++ interop.

## Parameters with reference types

Parameters that have mutable reference types are bridged as inout. Parameters with immutable reference types (const ref) are bridged as value types. ⚠ This will change as soon as Swift has a way to represent immutable borrows. ⚠

## Lifetimes

Currently, lifetimes are extended to the end of the lexical scope if any unsafe pointers are used in that scope. TODO: this should be updated to extend lifetimes whenever a C++ type is used in that scope. Currently, if there is no unsafe pointer used in the scope, then normal Swift lifetime rules apply.

## Borrowing Self

For mutating methods, self is borrowed and the access to self lasts for the duration of the call. For non-mutating methods, the access to self is currently instantaneous. ⚠ In the very near future we plan to borrow self in both cases. This will be a source breaking change from what native Swift methods do. ⚠

*More to come soon :)*