# Bridging to Objective-C

## ObjC to Swift

### Renaming

```
// ObjC
NS_SWIFT_NAME(Sandwich.Preferences)
@interface SandwichPreferences : NSObject
@property BOOL includesCrust NS_SWIFT_NAME(isCrusty);
@end
// Swift
var preferences = Sandwich.Preferences()
preferences.isCrusty = true


// Renaming an ObjC enum
typedef NS_ENUM(NSInteger, SandwichBreadType) {
    brioche, pumpernickel, pretzel, focaccia
} NS_SWIFT_NAME(SandwichPreferences.BreadType);
```

### Refining Method Names
- Initializer methods are imported by Swift with double underscores () prepended to their first argument labels.
- Object subscripting methods are imported by Swift as methods with double underscores () prepended to their base names, rather than as a Swift subscript, if either the getter or setter method is marked as NS_REFINED_FOR_SWIFT.
- Other methods are imported with double underscores () prepended to their base names.

```
// ObjC
@interface Color : NSObject
- (void)getRed:(nullable CGFloat *)red
         green:(nullable CGFloat *)green
          blue:(nullable CGFloat *)blue
         alpha:(nullable CGFloat *)alpha NS_REFINED_FOR_SWIFT;
```

```
@end
// Swift
extension Color {
    var rgba: (red: CGFloat, green: CGFloat, blue: CGFloat, alpha: CGFloat) {
        var r: CGFloat = 0.0
        var g: CGFloat = 0.0
        var b: CGFloat = 0.0
        var a: CGFloat = 0.0
        __getRed(red: &r, green: &g, blue: &b, alpha: &a)
        return (red: r, green: g, blue: b, alpha: a)
    }
}
```

## NSError and throws

NSFileManager:

```
// ObjC
- (BOOL)removeItemAtURL:(NSURL *)URL
            error:(NSError **)error;
// Swift
func removeItem(at: URL) throws
// Note the lack of a boolean return
```

- Use `NS_SWIFT_NOTHROW` to avoid the throws-based translation