

Objective-C Mapping for Modules

Because Objective-C does not support namespaces, a Slice module maps to a prefix for the identifiers defined inside the modules. By default, the prefix is the same as the name of the module:

Slice

```
module example
{
    enum Color { Red, Green, Blue };
};
```

With this definition, the Slice identifier `Color` maps to the Objective-C identifier `exampleColor`.

For nested modules, by default, the module identifiers are concatenated. For example, consider the following Slice definition:

Slice

```
module outer {
    module inner {
        enum Color { Red, Green, Blue };
    };
};
```

With this definition, the Slice identifier `Color` becomes `outerinnerColor` in Objective-C.

You can use a metadata directive to change the default mapping to a different prefix. For example:

Slice

```
["objc:prefix:OUT"]
module outer {
    enum Vehicle { Car, Truck, Bicycle };

    module inner {
        enum Color { Red, Green, Blue };
    };
};
```

With this definition, `Vehicle` maps to `OUTVehicle`. However, `Color` still maps to `outerinnerColor`, that is, the metadata directive applies only to types defined in the outer module, but not to types that are defined in nested modules. If you want to assign a prefix for types in the nested module, you must use a separate metadata directive, for example:

Slice

```
["objc:prefix:OUT"]
module outer {
    enum Vehicle { Car, Truck, Bicycle };

    ["objc:prefix:IN"]
    module inner {
        enum Color { Red, Green, Blue };
    };
};
```

With this definition, `Vehicle` maps to `OUTVehicle`, and `Color` maps to `INColor`.

For the remainder of the examples in this chapter, we assume that Slice definitions are enclosed by a module `Example` that is annotated with the metadata directive `["objc:prefix:EX"]`.

See Also

- [Objective-C Mapping for Identifiers](#)
- [Objective-C Mapping for Built-In Types](#)
- [Objective-C Mapping for Enumerations](#)
- [Objective-C Mapping for Structures](#)
- [Objective-C Mapping for Sequences](#)
- [Objective-C Mapping for Dictionaries](#)
- [Objective-C Mapping for Constants](#)
- [Objective-C Mapping for Exceptions](#)
- [Objective-C Mapping for Interfaces](#)