

PHP Mapping for Modules

By default, identifiers defined within a Slice [module](#) are mapped to a flattened symbol that uses underscores as module separators. Consider the following Slice definition:

Slice

```
module M {
    module N {
        enum Color { red, green, blue };
    };
};
```

The Slice identifier `Color` maps to `M_N_Color` by default because PHP releases prior to version 5.3 lacked language support for namespaces. If you prefer to use namespaces instead, you can enable an alternate mapping in which Slice modules map to PHP namespaces with the same name as the Slice module. This mapping preserves the nesting of the Slice definitions. Using the namespace mapping, the Slice identifier `Color` maps to `\M\N\Color`.

Be aware that using underscores in your Slice definitions can lead to name collisions in the flattened mapping. Consider the following example:

Slice

```
module M {
    module N {
        enum Color { red, green, blue };
    };
};

module M_N {
    interface Color { };
};
```

Although these definitions are syntactically correct, they both map to the flattened PHP symbol `M_N_Color`.

See Also

- [Modules](#)
- [PHP Mapping for Identifiers](#)
- [PHP Mapping for Built-In Types](#)
- [PHP Mapping for Enumerations](#)
- [PHP Mapping for Structures](#)
- [PHP Mapping for Sequences](#)
- [PHP Mapping for Dictionaries](#)
- [PHP Mapping for Constants](#)
- [PHP Mapping for Exceptions](#)