

Ruby Mapping for Identifiers



A Slice [identifier](#) maps to an identical Ruby identifier. For example, the Slice identifier `clock` becomes the Ruby identifier `clock`. There are two exceptions to this rule:

1. Ruby requires the names of classes, modules, and constants to begin with an upper case letter. If a Slice identifier maps to the name of a Ruby class, module, or constant, and the Slice identifier does not begin with an upper case letter, the mapping replaces the leading character with its upper case equivalent. For example, the Slice identifier `bankAccount` is mapped as `BankAccount`.
2. If a Slice identifier is the same as a Ruby keyword, the corresponding Ruby identifier is prefixed with an underscore. For example, the Slice identifier `while` is mapped as `_while`.



You should try to [avoid such identifiers](#) as much as possible.

[Back to Top ^](#)

See Also

- [Lexical Rules](#)
- [Ruby Mapping for Modules](#)
- [Ruby Mapping for Built-In Types](#)
- [Ruby Mapping for Enumerations](#)
- [Ruby Mapping for Structures](#)
- [Ruby Mapping for Sequences](#)
- [Ruby Mapping for Dictionaries](#)
- [Ruby Mapping for Constants](#)
- [Ruby Mapping for Exceptions](#)
- [Ruby Mapping for Interfaces](#)
- [Ruby Mapping for Operations](#)

