

Java Mapping for Identifiers

A Slice [identifier](#) maps to an identical Java identifier. For example, the Slice identifier `Clock` becomes the Java identifier `Clock`. There is one exception to this rule: if a Slice identifier is the same as a Java keyword or is an identifier reserved by the Ice run time (such as `checkedCast`), the corresponding Java 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.

A single Slice identifier often results in several Java identifiers. For example, for a Slice interface named `Foo`, the generated Java code uses the identifiers `Foo` and `FooPrx` (among others). If the interface has the name `while`, the generated identifiers are `_while` and `whilePrx` (*not* `_whilePrx`), that is, the underscore prefix is applied only to those generated identifiers that actually require it.

See Also

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