Ruby Mapping for Built-In Types

On this page:

- Mapping of Slice Built-In Types to Ruby Types
- String Mapping in Ruby

Mapping of Slice Built-In Types to Ruby Types

The Slice built-in types are mapped to Ruby types as shown in this table:

Slice	Ruby
`bool`	`true` or `false`
`byte`	`Fixnum`
`short`	`Fixnum`
`int`	`Fixnum` or `Bignum`
`long`	`Fixnum` or `Bignum`
`float`	`Float`
`double`	`Float`
`string`	`String`

Although Ruby supports arbitrary precision in its integer types, the Ice run time validates integer values to ensure they have valid ranges for their declared Slice types.

String Mapping in Ruby

String values returned as the result of a Slice operation (including return values, out parameters, and data members) contain UTF-8 encoded strings unless the program has installed a string converter, in which case string values use the converter's native encoding instead.

As string input values for a remote Slice operation, Ice accepts `nil` in addition to `String` objects; each occurrence of `nil` is marshaled as an empty string. Ice assumes that all `String` objects contain valid UTF-8 encoded strings unless the program has installed a string converter, in which case Ice assumes that `String` objects use the native encoding expected by the converter.

See Also

- Basic Types
- Ruby Mapping for Identifiers
- Ruby Mapping for Modules
- 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
- C++ Strings and Character Encoding

Copyright © 2017, ZeroC, Inc.