

PHP Mapping for Built-In Types

On this page:

- [Mapping of Slice Built-In Types to PHP Types](#)
- [String Mapping in PHP](#)

Mapping of Slice Built-In Types to PHP Types

PHP has a limited set of primitive types: `boolean`, `integer`, `double`, and `string`. The Slice [built-in](#) types are mapped to PHP types as shown in the table below:

Slice	Ruby
<code>bool</code>	<code>true</code> or <code>false</code>
<code>byte</code>	<code>integer</code>
<code>short</code>	<code>integer</code>
<code>int</code>	<code>integer</code>
<code>long</code>	<code>integer</code>
<code>float</code>	<code>double</code>
<code>double</code>	<code>double</code>
<code>string</code>	<code>string</code>

PHP's `integer` type may not accommodate the range of values supported by Slice's `long` type, therefore `long` values that are outside this range are mapped as strings. Scripts must be prepared to receive an `integer` or `string` from any operation that returns a `long` value.

String Mapping in PHP

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 `null` in addition to `string` objects; each occurrence of `null` 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](#)
- [PHP Mapping for Identifiers](#)
- [PHP Mapping for Modules](#)
- [PHP Mapping for Enumerations](#)
- [PHP Mapping for Structures](#)
- [PHP Mapping for Sequences](#)
- [PHP Mapping for Dictionaries](#)
- [PHP Mapping for Constants](#)
- [PHP Mapping for Exceptions](#)
- [C++ Strings and Character Encoding](#)