

PHP Mapping for Dictionaries

A Slice [dictionary](#) maps to a native PHP associative array. The PHP mapping does not currently support all Slice dictionary types, however, because native PHP associative arrays support only integers and strings as keys.

A Slice dictionary whose key type is an enumeration or one of the primitive types `boolean`, `byte`, `short`, `int`, or `long` is mapped as an associative array with an integer key.



Boolean values are treated as integers, with false equivalent to 0 (zero) and true equivalent to 1 (one).

A Slice dictionary with a `string` key type is mapped as an associative array with a string key. All other key types cause a warning to be generated.

Here is the definition of our [EmployeeMap](#):

Slice

```
dictionary<long, Employee> EmployeeMap;
```

You can create an instance of this dictionary as shown below:

PHP

```
$e1 = new Employee;
$e1->number = 42;
$e1->firstName = "Stan";
$e1->lastName = "Lipmann";

$e2 = new Employee;
$e2->number = 77;
$e2->firstName = "Herb";
$e2->lastName = "Sutter";

$em = array($e1->number => $e1, $e2->number => $e2);
```

The Ice run time validates the elements of a dictionary to ensure that they are compatible with the declared type; `InvalidArgumentException` exception is raised if an incompatible type is encountered.

See Also

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