PHP Mapping for Enumerations

PHP does not have an enumerated type, so a Slice enumeration is mapped to a PHP class: the name of the Slice enumeration becomes the name of the PHP class; for each enumerator, the class contains a constant with the same name as the enumerator. For example:

Slice

```
enum Fruit { Apple, Pear, Orange };
```

The generated PHP class looks as follows:

PHP

```
class Fruit
{
    const Apple = 0;
    const Pear = 1;
    const Orange = 2;
}
```

Suppose we modify the Slice definition to include a custom enumerator value:

Slice

```
enum Fruit { Apple, Pear = 3, Orange };
```

The generated PHP class changes accordingly:

PHP

```
class Fruit
{
    const Apple = 0;
    const Pear = 3;
    const Orange = 4;
}
```

Since enumerated values are mapped to integer constants, application code is not required to use the generated constants. When an enumerated value enters the Ice run time, Ice validates that the given integer is a valid value for the enumeration. However, to minimize the potential for defects in your code, we recommend using the generated constants instead of literal integers.

See Also

- Enumerations
- PHP Mapping for Identifiers
- PHP Mapping for Modules
- PHP Mapping for Built-In Types
- PHP Mapping for Structures
- PHP Mapping for Sequences
- PHP Mapping for Dictionaries
- PHP Mapping for Constants
- PHP Mapping for Exceptions