Client-Side Slice-to-PHP Mapping

The client-side Slice-to-PHP mapping defines how Slice data types are translated to PHP types, and how clients invoke operations, pass parameters, and handle errors. Much of the PHP mapping is intuitive. For example, Slice sequences map to PHP arrays, so there is essentially nothing new you have to learn in order to use Slice sequences in PHP.

Much of what appears in this chapter is reference material. We suggest that you skim the material on the initial reading and refer back to specific sections as needed. However, we recommend that you read at least the mappings for exceptions, interfaces, and operations in detail because these sections cover how to call operations from a client, pass parameters, and handle exceptions.

In order to use the PHP mapping, you should need no more than the Slice definition of your application and knowledge of the PHP mapping rules. In particular, looking through the generated code in order to discern how to use the PHP mapping is likely to be inefficient, due to the amount of detail. Of course, occasionally, you may want to refer to the generated code to confirm a detail of the mapping, but we recommend that you otherwise use the material presented here to see how to write your client-side code.

The `Ice` Module
All of the APIs for the Ice run time are nested in the `Ice` module, to avoid clashes with definitions for other libraries or applications. Some of the contents of the `Ice` module are generated from Slice definitions; other parts of the `Ice` module provide special-purpose definitions that do not have a corresponding Slice definition. We will incrementally cover the contents of the `Ice` module throughout the remainder of the manual.

A PHP application can load the Ice run time using the `require` statement:

```
require 'Ice.php';
```

If the statement executes without error, the Ice run time is loaded and available for use. You can determine the version of the Ice run time you have just loaded by calling the `stringVersion` function:

```
$icever = Ice_stringVersion();
```

Using the namespace mapping, you can refer to a global Ice function such as `stringVersion` either by its flattened name (as shown above) or by its namespace equivalent:

```
$icever = \Ice\stringVersion();
```

Topics