slice2php Command-Line Options

On this page:

`slice2php` Command-Line Options

The Slice-to-PHP compiler, `slice2php`, offers the following command-line options in addition to the standard options:

- `--all`
  Generate code for all Slice definitions, including those included by the main Slice file.

- `-n, --namespace`
  Generate code using PHP namespaces. Note that namespaces are only supported in PHP 5.3 or later. Also note that the Ice extension for PHP must be built with namespace support enabled.

- `--checksum`
  Generate checksums for Slice definitions.

Compiler Output in PHP

For each Slice file `X.ice`, `slice2php` generates PHP code into a file named `X.php` in the output directory. The default output directory is the current working directory, but a different directory can be specified using the `--output-dir` option.

Include Files in PHP

It is important to understand how `slice2php` handles include files. In the absence of the `--all` option, the compiler does not generate PHP code for Slice definitions in included files. Rather, the compiler translates Slice `#include` statements into PHP `require` statements in the following manner:

1. Determine the full pathname of the included file.
2. Create the shortest possible relative pathname for the included file by iterating over each of the include directories (specified using the `-I` option) and removing the leading directory from the included file if possible.
   For example, if the full pathname of an included file is `/opt/App/slice/OS/Process.ice`, and we specified the options `-I/opt/App` and `-I/opt/App/slice`, then the shortest relative pathname is `OS/Process.ice` after removing `/opt/App/slice`.
3. Replace the `.ice` extension with `.php`. Continuing our example from the previous step, the translated `require` statement becomes

   ```
   require "OS/Process.php";
   ```

   As a result, you can use `-I` options to tailor the `require` statements generated by the compiler in order to avoid absolute path names and match the organizational structure of your application's source files.

See Also

- Using the Slice Compilers