

Using the Slice Compilers

Ice provides a separate Slice compiler for each language mapping, as shown below:

Language	Compiler
C++	slice2cpp
Java	slice2java
C#	slice2cs
Objective-C	slice2objc
Python	slice2py
Ruby	slice2rb
PHP	slice2php

The Slice compilers.

The compilers share a similar command-line syntax:

```
<compiler-name> [options] file...
```

Regardless of which compiler you use, a number of command-line options are common to the compilers for any language mapping. (See the appropriate language mapping chapter for options that are specific to a particular language mapping.) The common command-line options are:

- `-h, --help`
Displays a help message.
- `-v, --version`
Displays the compiler version.
- `-DNAME`
Defines the preprocessor symbol *NAME*.
- `-DNAME=DEF`
Defines the preprocessor symbol *NAME* with the value *DEF*.
- `-UNAME`
Undefines the preprocessor symbol *{NAME}*.
- `-IDIR`
Add the directory *DIR* to the search path for `#include` directives.
- `-E`
Print the preprocessor output on `stdout`.
- `--output-dir DIR`
Place the generated files into directory *DIR*.
- `-d, --debug`
Print debug information showing the operation of the Slice parser.
- `--ice`
Permit use of the normally reserved prefix `Ice` for identifiers. Use this option only when compiling the source code for the Ice run time.
- `--underscore`
Permit use of underscores in Slice identifiers.

The Slice compilers permit you to compile more than a single source file, so you can compile several Slice definitions at once, for example:

```
slice2cpp -I. file1.ice file2.ice file3.ice
```

See Also

- [Slice Compilation](#)