

Language Mappings

The rules that govern how each Slice construct is translated into a specific programming language are known as *language mappings*. For example, for the [C++ mapping](#), a Slice sequence appears as an STL vector, whereas, for the [Java mapping](#), a Slice sequence appears as a Java array. In order to determine what the API for a specific Slice construct looks like, you only need the Slice definition and knowledge of the language mapping rules. The rules are simple and regular enough to make it unnecessary to read the generated code to work out how to use the generated API.

Of course, you are free to peruse the generated code. However, as a rule, that is inefficient because the generated code is not necessarily suitable for human consumption. We recommend that you familiarize yourself with the language mapping rules; that way, you can mostly ignore the generated code and need to refer to it only when you are interested in some specific detail.

Currently, Ice provides language mappings for C++, Java, C#, Python, Objective-C, and, for the client side, PHP and Ruby.

See Also

- [C++ Mapping](#)
- [Java Mapping](#)
- [C# Mapping](#)
- [Objective-C Mapping](#)
- [Python Mapping](#)
- [Ruby Mapping](#)
- [PHP Mapping](#)