

# Server-Side Slice-to-C++ Mapping

The mapping for Slice data types to C++ is identical on the client side and server side. This means that everything in [Client-Side Slice-to-C++ Mapping](#) also applies to the server side. However, for the server side, there are a few additional things you need to know — specifically how to:

- Initialize and finalize the server-side run time
- Implement servants
- Pass parameters and throw exceptions
- Create servants and register them with the Ice run time.

Because the mapping for Slice data types is identical for clients and servers, the server-side mapping only adds a few additional mechanisms to the client side: a small API to initialize and finalize the run time, plus a few rules for how to derive servant classes from skeletons and how to register servants with the server-side run time.

Although the examples we present are very simple, they accurately reflect the basics of writing an Ice server. Of course, for more sophisticated servers, you will be using [additional APIs](#), for example, to improve performance or scalability. However, these APIs are all described in Slice, so, to use these APIs, you need not learn any C++ mapping rules beyond those we describe here.

## Topics

- [The Server-Side main Function in C++](#)
- [Server-Side C++ Mapping for Interfaces](#)
- [Parameter Passing in C++](#)
- [Raising Exceptions in C++](#)
- [Object Incarnation in C++](#)
- [Asynchronous Method Dispatch \(AMD\) in C++](#)
- [Example of a File System Server in C++](#)