

# The Ice Run Time in Detail

This section describes in detail the server-side APIs of the Ice run time for synchronous, oneway, and datagram invocations. Communicators are the main handle to the Ice run time. They provide access to a number of run time resources and allow you to control the life cycle of a server. Object adapters provide a mapping between abstract Ice objects and concrete servants. Various implementation techniques are at your disposal to control the trade-off between performance and scalability; in particular, servant locators are a central mechanism that permits you to choose an implementation technique that matches the requirements of your application.

Ice provides both oneway and datagram invocations. These provide performance gains in situations where an application needs to provide numerous stateless updates. Batching such invocations permits you to increase performance even further.

The Ice logging mechanism is user extensible, so you can integrate Ice messages into arbitrary logging frameworks, and the `Ice::Stats` interface permits you to collect statistics for network bandwidth consumption.

Finally, even though Ice is location transparent, in the interest of efficiency, colocated invocations do not behave in all respects like remote invocations. You need to be aware of these differences, especially for applications that are sensitive to thread context.

## Topics

- [Communicators](#)
- [Communicator Initialization](#)
- [Object Adapters](#)
- [Object Identity](#)
- [The Current Object](#)
- [Servant Locators](#)
- [Default Servants](#)
- [Server Implementation Techniques](#)
- [Servant Evictors](#)
- [The Ice Threading Model](#)
- [Using Proxies](#)
- [Request Contexts](#)
- [Connection Timeouts](#)
- [Oneway Invocations](#)
- [Datagram Invocations](#)
- [Batched Invocations](#)
- [Locators](#)
- [Administrative Facility](#)
- [Logger Facility](#)
- [Stats Facility](#)
- [Location Transparency](#)
- [Automatic Retries](#)
- [Dispatch Interceptors](#)
- [C++ Strings and Character Encoding](#)
- [Plug-in Facility](#)
- [Custom Class Loaders](#)