Ice Services

The Ice core provides a sophisticated client-server platform for distributed application development. However, realistic applications usually require more than just a remoting capability: typically, you also need a way to start servers on demand, distribute proxies to clients, distribute asynchronous events, configure your application, distribute patches for an application, and so on.

Ice ships with a number of services that provide these and other features. The services are implemented as Ice servers to which your application acts as a client. None of the services use Ice-internal features that are hidden from application developers so, in theory, you could develop equivalent services yourself. However, having these services available as part of the platform allows you to focus on application development instead of having to build a lot of infrastructure first. Moreover, building such services is not a trivial effort, so it pays to know what is available and use it instead of reinventing your own wheel.

On this page:

Freeze and FreezeScript

Ice has a built-in object persistence service, known as *Freeze*. Freeze makes it easy to store object state in a database: you define the state stored by your objects in Slice, and the Freeze compiler generates code that stores and retrieves object state to and from a database. Freeze uses Berkeley DB as its database.

Ice also offers a tool set collectively called FreezeScript that makes it easier to maintain databases and to migrate the contents of existing databases to a new schema if the type definitions of objects change.

IceGrid Service

IceGrid is an implementation of an Ice location service that resolves the symbolic information in an indirect proxy to a protocol-address pair for indirect binding. A location service is only the beginning of IceGrid's capabilities.

IceGrid:

- allows you to register servers for automatic start-up: instead of requiring a server to be running at the time a client issues a request, IceGrid starts servers on demand, when the first client request arrives.
- provides tools that make it easy to configure complex applications containing several servers.
- supports replication and load-balancing.
- automates the distribution and patching of server executables and dependent files.
- provides a simple query service that allows clients to obtain proxies for objects they are interested in.

IceBox Server

IceBox is a simple application server that can orchestrate the starting and stopping of a number of application components. Application components can be deployed as a dynamic library instead of as a process. This reduces overall system load, for example, by allowing you to run several application components in a single Java virtual machine instead of having multiple processes, each with its own virtual machine.

IceStorm

IceStorm is a publish-subscribe service that decouples clients and servers. Fundamentally, IceStorm acts as a distribution switch for events. Publishers send events to the service, which, in turn, passes the events to subscribers. In this way, a single event published by a publisher can be sent to multiple subscribers. Events are categorized by topic, and subscribers specify the topics they are interested in. Only events that match a subscriber's topic are sent to that subscriber. The service permits selection of a number of quality-of-service criteria to allow applications to choose the appropriate trade-off between reliability and performance.

IceStorm is particularly useful if you have a need to distribute information to large numbers of application components. (A typical example is a stock ticker application with a large number of subscribers.) IceStorm decouples the publishers of information from subscribers and takes care of the redistribution of the published events. In addition, IceStorm can be run as a *federated* service, that is, multiple instances of the service can be run on different machines to spread the processing load over a number of CPUs.

IcePatch2

IcePatch2 is a software patching service. It allows you to easily distribute software updates to clients. Clients simply connect to the IcePatch2 server and request updates for a particular application. The service automatically checks the version of the client's software and downloads any updated application components in a compressed format to conserve bandwidth. Software patches can be secured using the Glacier2 service, so only authorized clients can download software updates.

IcePatch2 supersedes IcePatch, which was a previous version of this service.

Glacier2

Glacier2 is the Ice firewall traversal service: it allows clients and servers to securely communicate through a firewall without compromising security. Client-server traffic is SSL-encrypted using public key certificates and is bidirectional. Glacier2 offers support for mutual authentication as well as secure session management.

Glacier2 supersedes Glacier, which was a previous version of this service

See Also

- IceGrid
- Freeze
- FreezeScript
- Glacier2
- IceBox
- IceStorm
- IcePatch2