Starting the IceBox Server

Previous

Next

Incorporating everything we discussed previously, we can now configure and start IceBox servers.

On this page:

Starting the C++ IceBox Server

The configuration file for our example C++ service is shown below:

```
IceBox.Service.Hello=HelloService:create
Hello.Endpoints=tcp -p 10001
```

Notice that we define an endpoint for the object adapter created by the `Hello` service.

Assuming these properties reside in a configuration file named `config`, we can usually start the C++ IceBox server as follows:

C++11

```
icebox++11 --Ice.Config=config
```

C++98

```
icebox --Ice.Config=config
```

Additional command line options are supported, including those that allow the server to run as a Windows service or Unix daemon.

> ⓘ **32-bit IceBox on 64-bit Linux**
>
> On 64-bit Linux, the 32-bit IceBox executables (when provided) are named `icebox32` (C++98) and `icebox32++11` (C++11).

Back to Top ^

Starting the Java IceBox Server

Our Java configuration is nearly identical to the C++ version, except for the entry point specification:

```
IceBox.Service.Hello=HelloServiceI
Hello.Endpoints=tcp -p 10001
```

Notice that we define an endpoint for the object adapter created by the `Hello` service.

Assuming these properties reside in a configuration file named `config`, we can start the Java IceBox server as follows:

Java

```
$ java -jar icebox-3.7.0.jar --Ice.Config=config
```

Java Compat

```
$ java -jar icebox-compat-3.7.0.jar --Ice.Config=config
```

Back to Top ^

Starting the C# IceBox Server

The configuration file for our example C# service is shown below:

```
IceBox.Service.Hello=helloservice.dll:HelloService
Hello.Endpoints=tcp -p 10001
```

Notice that we define an endpoint for the object adapter created by the `Hello` service.

Assuming these properties reside in a configuration file named `config`, we can start the C# IceBox server as follows:

```
$ iceboxnet --Ice.Config=config
```

Back to Top ^

IceBox Server Failures

At startup, an IceBox server inspects its configuration for all properties having the prefix `IceBox.Service` and initializes each service. If initialization fails for a service, the IceBox server invokes the `stop` operation on any initialized services, reports an error, and terminates.

Back to Top ^

See Also

- Service Helper Class
- IceBox.*

Previous

Next