

# Standalone Deployment



The first step in creating a standalone deployment is to configure the [IceBox](#) server that hosts the IceFIX bridge. Let's assume that you want to host your IceFIX bridge on 192.168.2.10 and connect with a FIX acceptor named TP1 on `tp1.foo.com` at port 10000. In this example we'll store all configuration and database files in the directory `C:\TP1`.

Create an IceBox configuration file named `config.icebox`:

## config.icebox

```
IceBox.ServiceManager.Endpoints=tcp -h 192.168.2.10 -p 9998
IceBox.Service.TP1=IceFIXService,10:create --Ice.Config=config.tp1
```

The first line configures the endpoint for the IceBox `ServiceManager` object. This example uses port 9998 but you can pick any unused port on the host. The second line configures an instance of an IceFIX bridge as the IceBox service named `TP1`. (We assign the service name `TP1` to the bridge strictly as a convenience; it is not necessary for the service to have the same name as its FIX acceptor.) The configuration for the IceFIX bridge service is loaded from the file `config.tp1`.

We'll also create a configuration file for the `iceboxadmin` application named `config.iceboxadmin`:

## config.iceboxadmin

```
IceBoxAdmin.ServiceManager.Proxy=IceBox/ServiceManager:tcp -h 192.168.2.10 -p 9998
```

At a minimum, `config.tp1` must contain the following properties:

## config.tp1

```
TP1.Bridge.Endpoints=tcp -h 192.168.2.10 -p 12000
TP1.FIXConfig=C:\TP1\config.fix
Freeze.DbEnv.TP1.DbHome=C:\TP1\db
```

The first line configures the endpoint for the IceFIX bridge. In this case, we're using TCP and port 12000. Once again, you can choose any unused port. The second line configures the location of the FIX configuration file; here we are using `C:\TP1\config.fix`. The last property configures the location of the [Freeze](#) database directory, which contains the database files associated with the IceFIX bridge. Here the database files are stored in `C:\TP1\db`.

The bridge also requires a QuickFIX configuration file. This file must define only one initiator and include the remainder of the required configuration information. For example:

## config.fix

```
[DEFAULT]
ConnectionType=initiator
HeartBtInt=30
FileStorePath=C:\TP1\store
FileLogPath=log
StartTime=00:00:00
EndTime=00:00:00
UseDataDictionary=N
SocketConnectHost=tp1.foo.com
RefreshOnLogin=Y
PersistMessages=Y

[SESSION]
BeginString=FIX.4.2
SenderCompID=CLIENT1
TargetCompID=TP1
SocketConnectPort=10000
```

Before starting the IceFIX bridge for the first time, you must create the Freeze database directory:

```
> mkdir C:\TP1\db
```

To start the IceFIX bridge, run the `icebox` executable as follows:

```
> icebox --Ice.Config=C:\TP1\config.icebox
```

Next you must activate the bridge using the `icefixadmin` tool. This program requires a proxy for the bridge, which you can define in a configuration file. Let's define the property in a file named `config.admin`:

#### **config.admin**

```
IceFIXAdmin.Bridge=IceFIX/Bridge:tcp -h 192.168.2.10 -p 12000
```

Now we can start `icefixadmin`:

```
> icefixadmin --Ice.Config=C:\TP1\config.admin
```

The bridge starts in an inactive state and does not attempt to log into the FIX acceptor until the bridge is activated. You can determine the bridge's current status using the `status` command and then activate it using the `activate` command:

```
> icefixadmin --Ice.Config=C:\TP1\config.admin
Ice 3.4.2 Copyright 2003-2011 ZeroC, Inc.
>>> status
default
  status: not active
>>> activate
activating default
>>> status
default
  status: active
>>>
```

To assist you in diagnosing any issues that might arise while developing your IceFIX applications, we recommend that you configure the bridge to enable diagnostic messages by adding the following properties to `config.icefix`:

#### **config.icefix**

```
TP1.Trace.Incoming=1
TP1.Trace.Outgoing=1
TP1.Trace.Event=1
TP1.Trace.Bridge=1
```

For these changes to take effect, you must restart the bridge. Since the bridge is an IceBox service, you can use the `iceboxadmin` tool to stop the bridge and restart it:

```
> iceboxadmin --Ice.Config=C:\TP1\config.iceboxadmin stop TP1
> iceboxadmin --Ice.Config=C:\TP1\config.iceboxadmin start TP1
```

Once the bridge is restarted, you will begin to see much more diagnostic information. For example, starting and activating the bridge will show messages similar to those below:

```
> icebox --Ice.Config=C:\TP1\config.icebox
-- 12/06/11 01:52:27.558 icebox-TP1: Bridge: onCreate: FIX.4.2:CLIENT1->TP1
-- 12/06/11 01:52:27.589 icebox-TP1: FIX.4.2:CLIENT1->TP1: Created session
-- 12/06/11 01:52:56.871 icebox-TP1: Bridge: onCreate: FIX.4.2:CLIENT1->TP1
-- 12/06/11 01:52:56.902 icebox-TP1: FIX.4.2:CLIENT1->TP1: Created session
```

