IceIAP

IceIAP is a transport plug-in that enables clients to communicate via the Apple iAP protocol reserved for accessories.

On this page:

IceIAP Overview

IceIAP is an Ice plug-in that must be installed in your iOS client applications that need to communicate with accessories over Bluetooth, the Apple Lightning connector, or the Apple 30-pin connector. This section reviews some concepts that will help you as you learn more about IceIAP.

The IceIAP transport is based on Apple's External Accessory framework and enables Ice clients running on iOS devices to communicate with Ice servers running on connected accessories. This transport is a client-side only transport for iOS. It doesn't for instance provide the server-side transport that is required on the accessory side. For information on how to implement the server side, you need to be a MFI licensee and get in touch with ZeroC.

Accessory Discovery

An accessory can be discovered based on a number of attributes:

- its name
- its manufacturer
- its model number
- an advertised protocol

An accessory endpoint can be configured with any of these attributes to find an accessory.

Back to Top ^

Installing IceIAP

The IceIAP plug-in must be installed in every client that needs to communicates with accessories. The plug-in is only distributed as a static library named `IceIAP`. Your C++ or Objective-C clients should link with this library and register the plug-in with one of the following functions:

C++

```
Ice::registerIceIAP(bool loadOnInitialize = true)
```

ObjC

```
ICEregisterIceIAP(BOOL loadOnInitialize)
```

This function must be called before communicator initialization. The `loadOnInitialize` parameter specifies if the plug-in is installed when the communicator is initialized. If set to `false`, you will need to enable the plug-in by setting the `Ice.Plugin.IceIAP` property to `1`.

> `Ice::registerIceIAP` is a simple helper function that calls `Ice::registerPluginFactory`.

Refer to the next section for information on configuring the plug-in.

Back to Top ^

Using IceIAP

This section describes how to incorporate IceIAP into your Ice applications.

Proxy Endpoints

An iAP endpoint in a proxy specifies attributes that are used to find and connect to a matching accessory. An iAP endpoint has the following syntax:

```
iap [-p PROTOCOL] [-n NAME] [-m MANUFACTURER] [-o MODELNUMBER]
```

For example, to invoke on a proxy for the `hello` object running on an accessory that implements the `com.zeroc.helloWorld` protocol, use the following stringified proxy:

```
    hello:iap -p com.zeroc.helloWorld
```

To use the secured iAP endpoint, replace `iap` with `iaps`. You will also need to ensure that the IceSSL plug-in is loaded and configured when using `iaps` endpoints.

See Also

- Plug-in Facility
- IceSSL
- Proxy and Endpoint Syntax