Testing and Debugging XPC Code With an Anonymous Listener

This thread has been locked by a moderator.

643

IMPORTANT This post is now retired in favour of official documentation, namely TN3113 Testing and Debugging XPC Code With an Anonymous Listener. I'm leaving the original post here just for the record.

Testing and debugging XPC code can be tricky because there are two processes involved. Imagine you might have an app with an embedded XPC Service. To debug this you have to run two instances of the debugger, one connected to the app and another to the service, and then bounce between them. There is, however, an easier way, a way that allows you to test and debug all of your XPC code in a single process.

The trick is to set up an anonymous XPC listener. To start, tweak your XPC listener abstraction to accept an `NSXPCListener` instance. For example, say you have a `MyListener` class like this:

```
class MyListener {

    init() {
        self.listener = NSXPCListener.service()
    }

    let listener: NSXPCListener

    … more code …
}
```

Change the initialiser to look like this:

```
init(listener: NSXPCListener = .service()) {
    self.listener = listener
}
```

This uses the XPC Service's listener by default, but allows you to override that by passing a value to the `listener` parameter.

Now, in your test environment, call the `anonymousListener` method to create a anonymous listener and pass that to your listener abstraction:

```
let myListener = MyListener(listener: .anonymous())
```

On the client side, tweak your XPC connection abstraction to accept an `NSXPCConnection` instance. For example, say you have a `MyConnection` class like this:

```
class MyConnection {

    init() {
        self.connection = NSXPCConnection(serviceName: "com.example.MyService")
    }

    let connection: NSXPCConnection
}
```

Change the initialiser to look like this:

```
init(connection: NSXPCConnection = .init(serviceName: "com.example.MyService")) {
    self.connection = connection
}
```

This sets up a connection to the XPC Service's listener by default, but allows you to override that by passing a value to the `connection` parameter.

Finally, in your test environment, use the `init(listenerEndpoint:)` initialiser to create a connection to your anonymous listener:

```
let connection = NSXPCConnection(listenerEndpoint: myListener.listener.endpoint)
let myConnection = MyConnection(connection: connection)
```

You now have a connection connected to your listener, both running in the same process. This makes it much easier to debug your XPC code. It's also perfect for unit tests.

Share and Enjoy
—
Quinn "The Eskimo!" @ Developer Technical Support @ Apple
`let myEmail = "eskimo" + "1" + "@" + "apple.com"`

XPC

Reply

Posted 11 months ago by eskimo

Add a Comment

This site contains user submitted content, comments and opinions and is for informational purposes only. Apple disclaims any and all liability for the acts, omissions and conduct of any third parties in connection with or related to your use of the site. All postings and use of the content on this site are subject to the Apple Developer Forums Participation Agreement.

Copyright © 2022 Apple Inc. All rights reserved. Terms of Use | Privacy Policy | License Agreements