MFi Accessories

Developer Guide to Submitting Apps that work with Accessories

Version R2
February 21, 2011

General Requirements

This guide applies to developers creating iOS apps that communicate with external devices using custom protocols via the 30-pin dock connector or Bluetooth.

Custom protocols are implemented in iOS apps using the External Accessory framework and in devices using the iPod Accessory Protocol.

Developers creating iOS apps must participate in the *iOS Developer Program*. Developers creating devices using the iPod Accessory Protocol must participate in the *MFi Program*. The same developer may create both iOS apps and devices by participating in both programs.

This guide does ***not*** apply to:

- iOS apps that communicate with external devices via Wi-Fi or other TCP/IP-based network protocols
- iOS apps that communicate with external devices using standard protocols such as Core Location, Core Audio or UIEvent remote control commands

Steps for the device developer

Device developers must meet all *MFi Program* requirements. The following steps are required for devices that communicate with iOS apps.

1. Add all iOS apps that work with your device to your MFi Product Plan.

   iOS apps not listed on your MFi Product Plan will be rejected if submitted to the App Store. The following information must be provided for each iOS app:

   - name
   - version number
   - planned release date
   - App Store category
   - bundle identifier
   - protocol names
   - functional overview
   - name of the developer that will submit the app to the App Store

2. Program your device firmware to declare the preferred app and a list of supported protocol names.

   The preferred app is specified using `SetFIDTokenValues`, `BundleSeedIDPrefToken` and must match the Bundle Seed ID of the preferred app. The Bundle Seed ID is a unique ten digit number assigned to app developers in the *iOS Provisioning Portal*. If you are not the app developer, you will need to obtain the Bundle Seed ID from the app developer.

   Supported protocol names are specified using `SetFIDTokenValues`, `EAProtocolToken` and must match one or more protocol names in the app's `Info.plist`.

3. Complete MFi certification.

   All steps up to and including production certification must be complete ***before*** the iOS app is submitted for review.

Steps for the app developer

App developers must meet all *iOS Developer Program* requirements. The following steps are required for apps that communicate with devices.

1. Specify protocol names that match the target device.

   The `UISupportedExternalAccessoryProtocols` key in your app’s `Info.plist` must contain one or more protocol names that match protocol names declared by the device.

   Your app must identify correct protocol names when enumerating connected devices using the External Accessory framework.

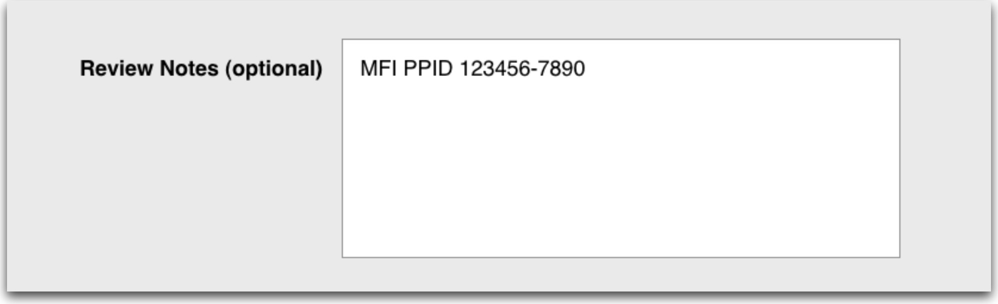
2. Confirm that the target device has completed MFi certification.

   All steps up to and including production certification must be complete ***before*** you submit your iOS app for review.

3. Provide the unique MFI PPID for the target device in *iTunes Connect*.

   When you submit your app using *iTunes Connect*, you must enter the device’s unique MFI PPID in your app metadata *Review Notes* field. The device’s unique MFI PPID is a ten digit number that is assigned when the device’s Product Plan is created. If you are not the device developer, you will need to obtain the MFI PPID from the device developer.

Example



Review Notes (optional) MFI PPID 123456-7890