Adding the MobileFirst Platform Foundation SDK to iOS Applications

Overview

The MobileFirst Platform Foundation SDK consists of a collection of pods, available through CocoaPods (http://guides.cocoapods.org), that you can add to your Xcode project. The pods correspond to core functions and other functions:

- **IBMMobileFirstPlatformFoundation** - Implements client/server connectivity, handles authentication and security aspects, resource requests, and other required core functions.
- **IBMMobileFirstPlatformFoundationJSONStore** - Contains the JSONStore framework. For more information, review the JSONStore for iOS tutorial (../../using-the-mfpf-sdk/jsonstore-ios/).
- **IBMMobileFirstPlatformFoundationPush** - Contains the Push Notifications framework. For more information, review the Notifications tutorials (../../notifications/).
- **IBMMobileFirstPlatformFoundationWatchOS** - Contains support for the Apple WatchOS. For more information, review the user documentation (http://www-01.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/com.ibm.worklight.dev.doc/devref/t_ios_frameworks.html).

In this tutorial you will learn how to add the MobileFirst Native SDK using CocoaPods to either a new or existing iOS application. You will also learn how to configure the MobileFirst Server to recognize the application, as well as find information about the MobileFirst configuration files that are added to the project.

> For instruction to manually add the SDK files to a project, visit the user documentation (http://www-01.ibm.com/support/knowledgecenter/SSHS8R_8.0.0/wl_welcome.html).

Prerequisites:

- Xcode and MobileFirst Developer CLI installed on the developer workstation.
- MobileFirst Server to run locally, or a remotely running MobileFirst Server.
- Make sure you have read the Setting up your MobileFirst development environment (../../setting-up-your-development-environment/mobilefirst-development-environment) and Setting up your iOS development environment (../../setting-up-your-development-environment/ios-development-environment) tutorials.

Jump to:

- Adding the MobileFirst Native SDK
- Updating the MobileFirst Native SDK
- Generated MobileFirst Native SDK artifacts
- Tutorials to follow next

Adding the MobileFirst Native SDK

Follow the below instructions to add the MobileFirst Native SDK to either a new or existing Xcode project, and register the application in the MobileFirst Server.

Before starting, make sure the MobileFirst Server is running.
If using a locally installed server: From a **Command-line** window, navigate to the server's **scripts** folder and run the command: `./start.sh`.

Creating an application

Create an Xcode project or use an existing one (Swift or Objective-C).

Adding the SDK

1. The MobileFirst Native SDK is provided via CocoaPods. If CocoaPods (http://guides.cocoapods.org) is not installed in your development environment, install it as follows:
   - Open a **Command-line** window and navigate to the root of the Xcode project.
   - Run the command: `sudo gem install cocoapods` followed by `pod setup`. **Note:** These commands may take several minutes to complete.
2. Run the command: `pod init`. This creates a `Podfile`.
   - Using your favorite code editor, open the `Podfile`.
   - Comment out or delete the contents of the file.
   - Add the following lines and save the changes:

```
# remove source line below before going live with the tutorials
source 'https://hub.jazz.net/git/oper2000/imf-client-sdk-specs-inhouse.git'
use_frameworks!
pod 'IBMMobileFirstPlatformFoundation'
```

3. Run the command: `pod install`. This command adds the MobileFirst Native SDK files, adds the **mfpclient.plist** file and generates a Pod project.
   Note: This command may take several minutes to complete.

   > **Important**: From here on, use the `[ProjectName].xcworkspace` file in order to open the project in Xcode. Do **not** use the `[ProjectName].xcodeproj` file. A CocoaPods-based project is managed as a workspace containing the application (the executable) and the library (all project dependencies that are pulled by the CocoaPods manager).

Registering the application

1. Open a **Command-line** window and navigate to the root of the Xcode project.
2. Run the command:

   ```
   mfpdev app register
   ```

   You will be asked to provide the application's BundleID.
   The BundleID is **case sensitive***.
   The `mfpdev app register` CLI command first connects to the MobileFirst Server to register the application, followed by generating the **mfpclient.plist** file at the root of the Xcode project, and adding to it the metadata that identifies the MobileFirst Server.

   > **Tip:** The application registration can also be performed from the MobileFirst Operations Console:
   > 1. Open your browser of choice and load the MobileFirst Operations Console using the address `http://localhost:9080/mfpconsole/`. You can also open the console from the **Command-line** using the CLI command `mfpdev server console`.
   > 2. Click on the "New" button next to "Applications" to create a new application and follow the on-screen instructions.
   > 3. After successfully registering your application you can optionally download a "skeleton" Xcode project pre-bundled with the MobileFirst Native SDK.

Completing the setup process

In Xcode, right-click the project entry, click on **Add Files To [ProjectName]** and select the **mfpclient.plist** file, located at the root of the Xcode project.

Referencing the SDK

Whenever you want to use the MobileFirst Native SDK, make sure that you import the MobileFirst Platform Foundation framework:

Objective-C:

```
#import <IBMMobileFirstPlatformFoundation/IBMMobileFirstPlatformFoundation.h>
```

Swift:

```
import IBMMobileFirstPlatformFoundation
```

Note about iOS 9:

- If you are developing for iOS 9, consider disabling ATS (http://iosdevtips.co/post/121756573323/ios-9-xcode-7-http-connect-server-error) in the application's `info.plist` to be able to test locally without security restrictions.
- Xcode 7 enables Application Transport Security (ATS) (https://developer.apple.com/library/ios/releasenotes/General/WhatsNewIniOS/Articles/iOS9.html#//apple_ref/doc/uid/TP40016198-SW14) by default.
  To complete the tutorial, disable ATS (http://iosdevtips.co/post/121756573323/ios-9-xcode-7-http-connect-server-error).

  1. In Xcode, right-click the **[project]/info.plist file → Open As → Source Code**
  2. Paste the following:

     ```
     <key>NSAppTransportSecurity</key>
     <dict>
       <key>NSAllowsArbitraryLoads</key>
       <true/>
     </dict>
     ```

Updating the MobileFirst Native SDK

To update the MobileFirst Native SDK with the latest release, run the following command from the root folder of the Xcode project in a **Command-line** window:

```
pod update
```

SDK releases can be found in the SDK's CocoaPods repository (https://cocoapods.org/?q=ibm%20mobilefirst).

Generated MobileFirst Native SDK artifacts

mfpclient.plist

Located at the root of the project, this file contains server connectivity properties and is user-editable:

- `protocol` – The communication protocol to MobileFirst Server. Either `HTTP` or `HTTPS`.
- `host` – The hostname of the MobileFirst Server instance.
- `port` – The port of the MobileFirst Server instance.
- `wlServerContext` – The context root path of the application on the MobileFirst Server instance.
- `languagePreference` - Sets the default language for client sdk system messages

Tutorials to follow next

With the MobileFirst Native SDK now integrated, you can now:

- Review the Adapters development tutorials (../../adapters/)
- Review the Authentication and security tutorials (../../authentication-and-security/)
- Review the Notifications tutorials (../../notifications/)
- Review All Tutorials (../../all-tutorials)