Push Notifications in Hybrid Applications

Overview

This tutorial explains how to configure a MobileFirst hybrid application to support push notifications. Also mentioned are the addresses and ports that are required to be accessible for notifications to arrive to/from the supported Push Notification Service vendors (APNS, GCM, WNS, and MPNS).

Agenda

- Setup and guidelines
  - iOS
  - Android
  - Windows Phone 8
  - Windows 8
- Notification types

Setup and guidelines

Push notification support is set up differently depending on the platform:

iOS

To send push notifications to iOS devices, you use the Apple Push Notifications Service (APNS).
You must be a registered Apple iOS Developer to obtain an APNS certificate for your application.

- During the development phase, use the `apns-certificate-sandbox.p12` sandbox certificate file.
- During the production phase, use the `apns-certificate-production.p12` production certificate file.
- Place the certificate file either in the application root folder or in the application environment (iPhone or iPad) folder. The environment root folder takes the highest priority.

Note: APNS certificates must have a non-blank password.

Note: The APNS production certificate can only be tested once the application that utilizes it has been successfully submitted to the Apple App Store.

When the hybrid application has both iPhone and iPad environments, it requires separate certificates for push notification for each environment. In that case, place those certificates in the corresponding environment folders.

> To learn about setting up push notifications required certificates, see this blog post: Understanding and setting up certificates required to use iOS devices and Apple Push Notifications services (APNS) (https://www.ibm.com/developerworks/community/blogs/worklight/entry/understanding-and-setting-up-push-notifications-in-development-evnironment?lang=en)

Apple Push Notifications Service

For push notifications to be sent, the following servers must be accessible from a MobileFirst Server

instance:
Sandbox servers:
gateway.sandbox.push.apple.com:2195
feedback.sandbox.push.apple.com:2196

Production servers:
gateway.push.apple.com:2195
Feedback.push.apple.com:2196

1-courier.push.apple.com 5223

Project configuration

1. Place the APNS certificate file at the root of the application folder or at the root of the environment folder.
2. In the `application-descriptor.xml` file, add the `pushSender` element under the iphone environment. You can also edit these settings with the Application Descriptor Editor in Design mode.
    - Replace the `password` value with the .p12 certificate password.
    - Replace the `bundleId` value with your application `bundleId`. Consult the Apple documentation about how to create a `bundleId` for Xcode projects.

```
<iphone bundleId="com.REPLACE-WITH-BUNDLE-ID" version="1.0">
   <worklightSettings include="false"/>
   <pushSender password="REPLACE-WITH-CERTIFICATE-PASSWORD" />
   <security>
      <encryptWebResources enabled="false"/>
      <testWebResourcesChecksum enabled="false" ignoreFileExtensions="png, jpg, jpeg, gif
, mp4, mp3"/>
   </security>
</iphone>
```

Android

To send push notifications to Android devices, you use the Google Cloud Messaging (GCM) service.
To register to the GCM service, you need a valid Gmail account.

For more information about how to get a GCM Project Number and API key (which you use later in the MobileFirst project), see the Google Developer Console (https://console.developers.google.com).

Notes:

- When you create an API key, make sure that the type is **server key**.
- Android OS 2.3.5 devices must be synchronized with a regular Gmail account.
- Android OS 4.x devices do not impose Gmail account synchronization.

Project configuration

To set up push notifications, edit the `application-descriptor.xml` file. Add the `pushSender` element under the Android environment. You can also edit these settings with the Application Descriptor Editor in Design mode.

```
<android version="1.0">
  <worklightSettings include="false"/>
  <pushSender key="REPLACE-WITH-API-KEY-VALUE" senderId="REPLACE-WITH-PROJECT-NUMBER" />
  <security>
    <encryptWebResources enabled="false"/>
    <testWebResourcesChecksum enabled="false" ignoreFileExtensions="png, jpg, jpeg, gif, mp4, mp3"/>
    <publicSigningKey/>
    <packageName/>
  </security>
</android>
```

Use the values that you previously created in the GCM website:

- Replace `key` value with the **API Key** value.
- Replace `senderId` value with the **Project Number** value.

Add Google Play Services

For instructions about how to setup Google Play Services, see the Setting Up Google Play Services (http://developer.android.com/google/play-services/setup.html) topic at the Android Developer website.

Android Push Notifications Service

If your organization has a firewall that restricts the traffic to or from the Internet, you must go through the following steps:

- Configure the firewall to allow connectivity with GCM in order for your GCM client apps to receive messages. The ports to open are 5228, 5229, and 5230. GCM typically uses only 5228, but it sometimes uses 5229 and 5230. GCM does not provide specific IP, so you must allow your firewall to accept outgoing connections to all IP addresses contained in the IP blocks listed in Google's ASN of 15169. For more information, see Implementing an HTTP Connection Server (https://developers.google.com/cloud-messaging/http).
- Ensure that your firewall accepts outgoing connections from MobileFirst Server to `android.googleapis.com` on port 443.

Windows Phone 8

To send push notifications to Windows Phone 8 devices, you use the Microsoft Push Notifications Service (MPNS).

- Non-authenticated push notification does not require any setup from the developer. Authenticated push notification requires a Windows Phone Dev Center account.
- To use authenticated push, you must use a certificate that is issued by a Microsoft-trusted root certificate authority. *For production, consider using authenticated push notification to ensure that the information is not compromised.*

Windows Phone 8 Push Notifications Service

No specific port needs to be open in your server configuration.
MPNS uses regular http or https requests.

Project configuration

To set up push notifications, edit the `application-descriptor.xml` file. Add the `pushSender` element under the windowsPhone8 environment. You can also edit these settings with the Application Descriptor Editor in Design mode.

- Non-authenticated push

```
<windowsPhone8 version="1.0">
   <uuid>auto-generated by the platform</uuid
>
   <pushSender />
</windowsPhone8>
```

- Authenticated push

```
<windowsPhone8 version="1.0">
   <uuid>auto-generated by the platform</uuid>
   <pushSender>
      <authenticatedPush serviceName="" keyAlias="" keyAliasPassword=""/
>
   </pushSender>
</windowsPhone8>
```

  - Replace **`serviceName`** value with the service name.
  - Replace **`keyAlias`** value with the certificate alias.
  - Replace **`keyAliasPassword`** value with the certificate password.

> For more information about using the certificate file, see the topic about setting up push notifications for Windows Phone 8, in the user documentation.

Windows 8

To send push notifications to Windows 8 devices, the Windows Push Notification Service (WNS) is used. As a developer, you need to register your app with Windows Store through the Windows Dev Center by using your Microsoft account.

For more information about how to get WNS credentials (which you will use later in the MobileFirst project), see http://msdn.microsoft.com/en-in/library/windows/apps/hh465407.aspx (http://msdn.microsoft.com/en-in/library/windows/apps/hh465407.aspx)

Windows 8 Push Notifications Service

No specific port needs to be open in your server configuration.
WNS uses regular http or https requests.

Project configuration

To set up push notifications, edit the `application-descriptor.xml` file. Add the `pushSender` element under the windows8 environment. You can also edit these settings with the Application Descriptor Editor in Design mode.

```
<windows8 version="1.0">
   <uuid>auto-generated by the platform</uuid>
   <pushSender clientSecret="" packageSID="" appIdentityName="" appIdentityPublisher=""/
>
</windows8>
```

Use the values that you previously generated in the Windows Store Dashboard:

- Replace `clientSecret` value with the secret key.
- Replace `packageSID` value with the package security identifier.
- Replace `appIdentityName` value with the package name.
- Replace `appIdentityPublisher` value with the subject name of the certificate.

Notification Types