Application Authenticity Protection in Hybrid applications

This is a continuation of the Application Authenticity Protection (../) tutorial.

application-descriptor.xml

Add the *securityTest* attribute to the relevant environment element. For example:

```
<iphone bundleId="com.worklight.MyBankApp" applicationId="MyBankApp" securityTest="customTests"
version="1.0">
```

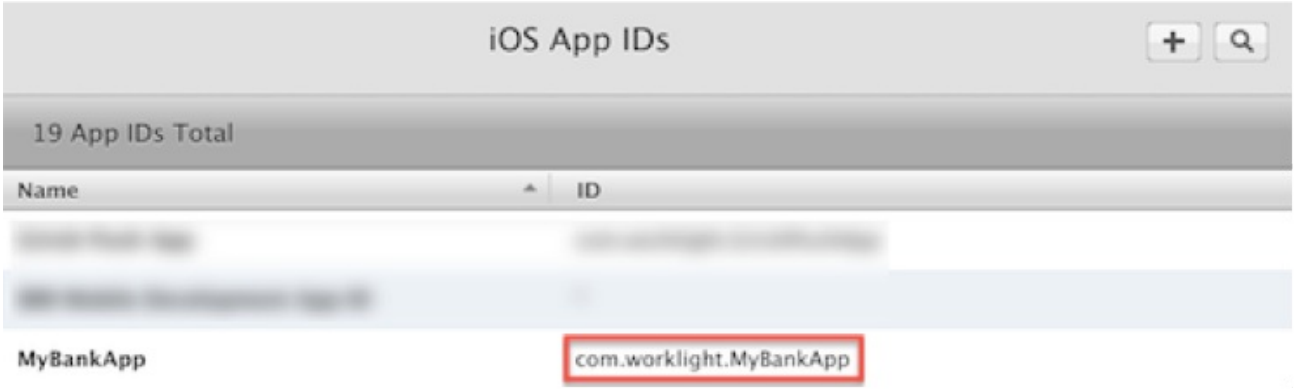
Next, specific environment modifications are required as well.

iOS

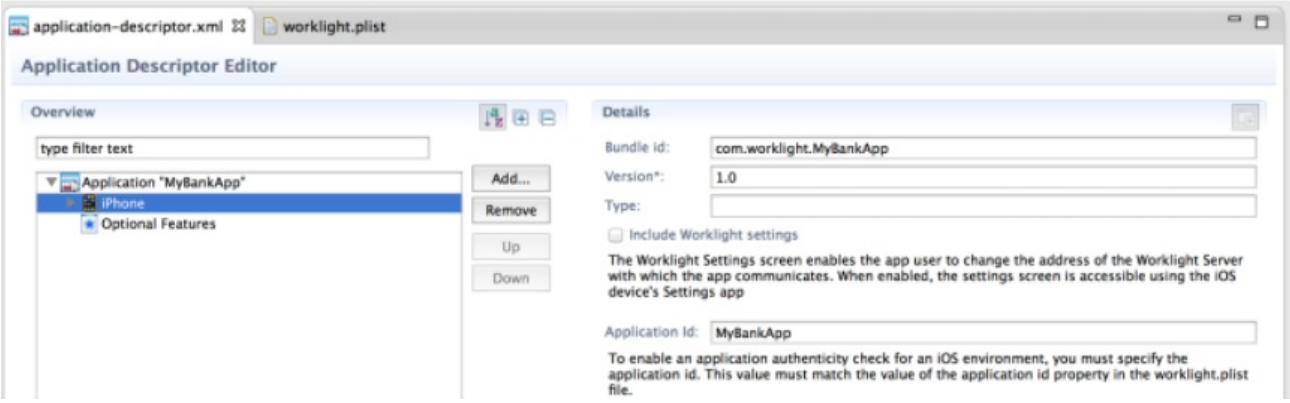
To enable application authenticity protection check for the iPhone/iPad environment, specify the following in `application-descriptor.xml`:

Specifying the bundleId and applicationId

1. Specify the `bundleId` of the application exactly as it was defined in the **Apple Developer portal**.



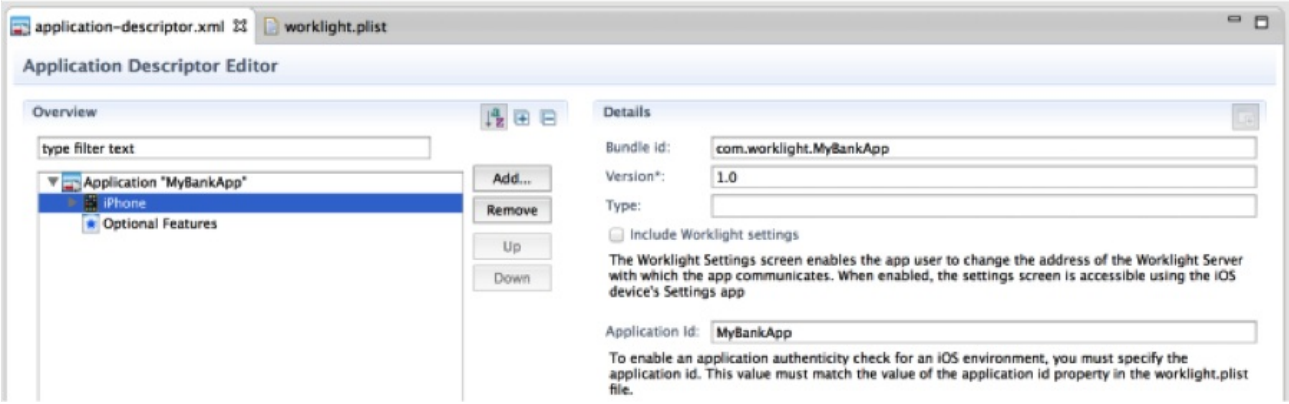
It can be added either in the Application Description Design view:



Or in the Application Descriptor Source view:

```
<iphone bundleId="com.worklight.MyBankApp" version="1.0">
```

2. Specify the `applicationId` value. The Application Id value must match the value of the `application id` property, which is located in the `native\worklight.plist` file.It can be added either in the Application Description Design view:



Or in the Application Descriptor Source view:

```
<iphone bundleId="com.worklight.MyBankApp" applicationId="MyBankApp" securityTest="custom
Tests" version="1.0">
```

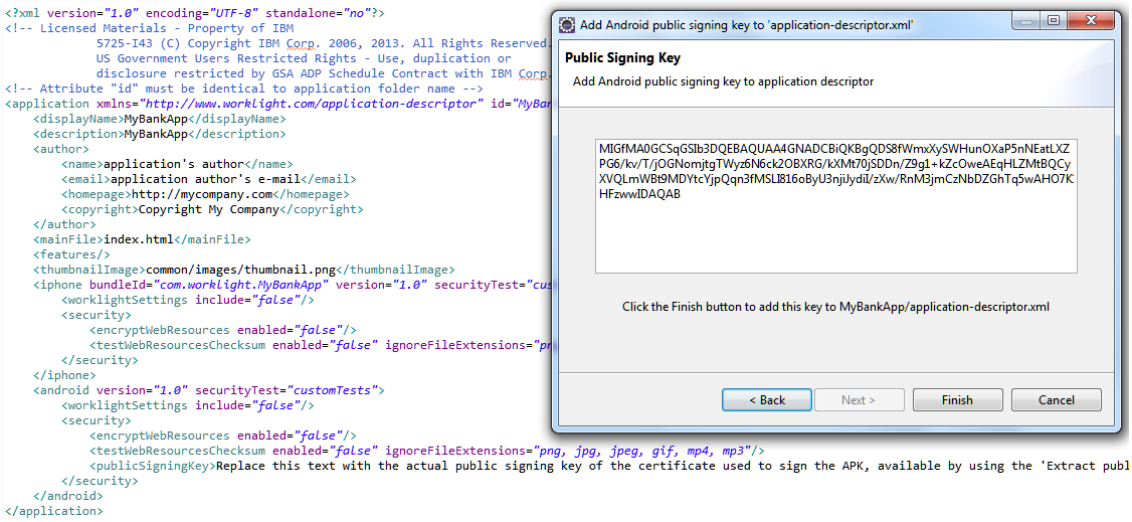
3. In Xcode, verify that the following value exists in the **Other Linker Flags** field: `-ObjC`

Android

To enable application authenticity protection check for the Android Hybrid environment:

1. Extract the public signing key of the certificate that is used to sign application bundle (*.apk* file).
   - If building the application for distribution (production), extract the public key from the certificate that is used to sign the production ready application.
   - If building an application in the development environment, the default public key that is supplied by the Android SDK can be used. The development certificate can be found in a keystore that is in a **{user-home}/.android/debug.keystore** file.
   - The public signing key can be extracted either manually or by using the wizard that MobileFirst Studio provides.
     To use the wizard:
     - Right-click the Android environment folder and select **Extract public signing key**.
     - Specify the location and the password of a keystore file, and click **Load Keystore**.
     - The default password for **debug.keystore** is "android".
     - Set the **Key alias** and click **Next**.
     - A dialog opens that displays the public key.
     - After you click **Finish**, the public key is automatically pasted to the relevant section of the

application-descriptor.xml file.



2. Add the Application package name by using the Application Descriptor Editor (design view):

3. Take the Application package name value from the package attribute of the *manifest* node in the **AndroidManifest.xml**.

   If you decide to change the value to another, verify that you change it in both locations.You can also directly edit **application-descriptor.xml** and add a *packageName*:

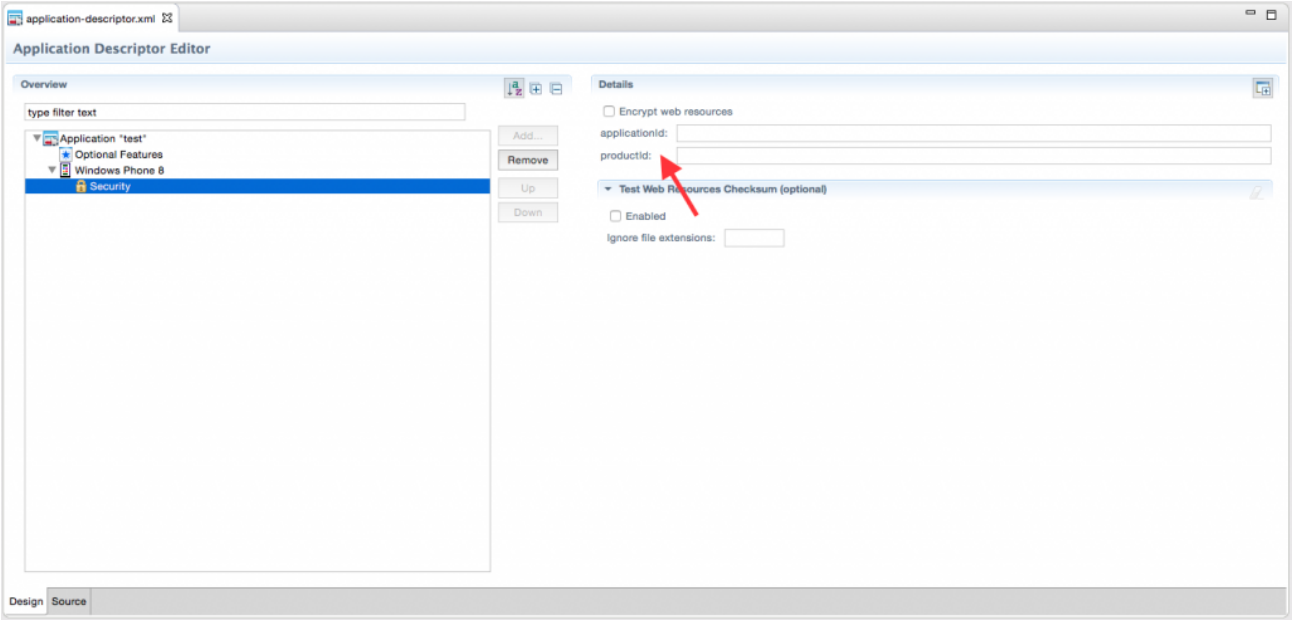
```
<android version="1.0">
  <worklightSettings include="false"/>
  <security>
    <encryptWebResources enabled="false"/>
    <testWebResourcesChecksum enabled="false" ignoreFileExtensions="png, jpg, jpeg, gif, mp4, mp3"/>
    <publicSigningKey>MIGff ...</publicSigningKey>
    <packageName>com.MyBankApp</packageName>
  </security>
</android>
```

Windows Phone 8

To enable application authenticity check for the Windows Phone 8 Hybrid environment, several modifications to the `application-descriptor.xml` file are needed:

1. In the Application Descriptor Design view, supply the `applicationId` and `productId` in the

Windows Phone 8 Security section:



2. The `productId` value can be found in `native\PropertiesWMAppManifest.xml`.
3. The `applicationId` value must match the value of the span style="font-family:courier-new">wlAppId property, which can be found in the span style="font-family:courier-new">native\wlclient.properties file.

These values can also be supplied in the Application Descriptor Source view. For example:

```
<windowsPhone8 version="1.0">
  <uuid>b5542877-7afe-4edc-a817-5341b5027633</uuid>
  <security>
    <productId>fca81480-7b4a-4ed0-b387-078e8fa0c3d5</productId
>
    <applicationId>HelloWorld</applicationId>
  </security>
</windowsPhone8>
```