# Table of Contents

1. Preface ........................................................................................................................................... 4
   1.1 Intended Audience ...................................................................................................................... 4
   1.2 Documentation Accessibility ...................................................................................................... 4
   1.3 Access to Oracle Support ........................................................................................................... 4
   1.4 Structure .................................................................................................................................... 4
   1.5 Related Information Sources ....................................................................................................... 4

2. OBDX Servicing Application .......................................................................................................... 5
   2.1 Pre requisite ............................................................................................................................... 5
   2.2 Create Project ............................................................................................................................. 5
   2.3 Adding UI to workspace. ............................................................................................................. 6
   2.4 Open project in Xcode ................................................................................................................ 6
   2.5 Generating Certificates for Development, Production and Push Notifications ......................... 7

3. Archive and Export .......................................................................................................................... 11

4. OBDX Authenticator Application .................................................................................................... 14
   4.1 Building Authenticator UI .......................................................................................................... 14
   4.2 Authenticator Application Workspace Setup .............................................................................. 16
   4.3 Building Authenticator Application ........................................................................................... 18
1. **Preface**

1.1 **Intended Audience**

This document is intended for the following audience:

- Customers
- Partners

1.2 **Documentation Accessibility**

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=accandid=docacc.

1.3 **Access to Oracle Support**

Oracle customers have access to electronic support through My Oracle Support. For information, visit

http://www.oracle.com/pls/topic/lookup?ctx=accandid=info or visit

http://www.oracle.com/pls/topic/lookup?ctx=accandid=trs if you are hearing impaired.

1.4 **Structure**

This manual is organized into the following categories:

*Preface* gives information on the intended audience. It also describes the overall structure of the User Manual.

The subsequent chapters describes following details:

- Configuration / Installation.

1.5 **Related Information Sources**

For more information on Oracle Banking Digital Experience Release 18.1.0.0.0, refer to the following documents:

- Oracle Banking Digital Experience Licensing Guide
2. OBDX Servicing Application

2.1 Pre requisite

- Download and Install node js as it is required to run npm and cordova commands.
- XCode 10 to be download from Mac App Store

2.2 Create Project

Ensure **Nodejs Version is >= 8 and Xcode version is >= 9.4.1**

Run Following commands on **terminal**

1. Install cordova using the command
   
   ```
   sudo npm install –g cordova
   ```
   
   a. If you face the above error then set proxy using following commands on command line.
      
      ```
      npm config set proxy <provide your proxy value here>
      npm config set https-proxy <provide your proxy value here>
      ```
   
   2. Create project using following command
      
      ```
      cordova create <directory name> <package name / app identifier> <project name>
      ```
      
      Eg : cordova create zigbank com.ofss.zigbank ZigBank
   
   3. All subsequent commands need to be run within the project’s directory
      
      ```
      cd <directory name>
      ```
      
      Eg: cd zigbank
   
   4. Add platform ios to the project using following command
      
      ```
      cordova platform add ios@4.x.x
      ```
5. Extract iOS workspace from installer and place in a folder.
   a. Copy folders from previously created sample project - cordova & CordovaLib (zigbank/platforms/ios) to this workspace at zigbank/platforms/ios/ios.

6. The workspace by default contains framework for running on devices. Hence to run the application on simulator, delete and copy the 2 frameworks (OBDXExtensions.framework, OBDXFramework.framework) from installer/simulator to zigbank/platforms/ios directory.

2.3 Adding UI to workspace.

   Use any 1 option below
   a. Building un built UI (required in case of customizations)

      Extract unbuilt UI and traverse to OBDX_Installer/installables/ui/channel/_build folder and perform below steps

      Windows –

      ```
      npm install -g grunt-cli
      npm install
      set IS_GRUNT=true
      node render-requirejs/render-requirejs.js mobile
      grunt --max_old_space_size=5120 mobilebuild --platform=ios
      ```

      Linux -

      ```
      sudo npm install -g grunt-cli
      sudo npm install
      export IS_GRUNT=true
      node render-requirejs/render-requirejs.js mobile
      grunt --max_old_space_size=5120 grunt mobilebuild --platform=ios
      ```

   b. Using built UI (out of box shipped with installer)

      i. Unzip dist.tar.gz for android from installer and copy folders(build.fingerprint,components,corporate,extensions,framework,images,index.html,manifest.json,pages,partials,resources,retail,sw.js) to workspace (platforms/ios/www/)

2.4 Open project in Xcode

   Open Xcode by clicking ZigBank.xcodeproj at zigbank/platforms/ios/

   1. Adding URLs to app.plist (ZigBank/Resources)

      a. FOR NONOAM (DB Authenticator setup)

      | LoginController          | NONOAM                  |
      |--------------------------|-------------------------|
      | KEY_SERVER_URL           | Eg http://mum00cag.in.oracle.com:7780 |
b. OAM Setup (Refer to installer pre requisite documents for OAuth configurations)

<table>
<thead>
<tr>
<th>LoginController</th>
<th>OAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEY_OAM_URL</td>
<td>Eg. <a href="http://mum00aoo.in.oracle.com:14100">http://mum00aoo.in.oracle.com:14100</a> (This URL must be of OHS without webgate)</td>
</tr>
<tr>
<td>KEY_SERVER_URL</td>
<td>Eg.<a href="http://mum00cag.in.oracle.com:7780">http://mum00cag.in.oracle.com:7780</a></td>
</tr>
</tbody>
</table>

2.5 Generating Certificates for Development, Production and Push Notifications

Create all certificates (by uploading CSR for keychain utility), provisioning profiles and push certificates as shown below by login in developer console. For development add device UUIDs and add same to provisioning profiles. Add capabilities as shown below and ensure the bundle identifier matches the one of the application in Xcode.
Ensure AppGroups capability is added to all profiles and for mobile profile SiriKit, App Groups, Push Notifications must be added.

Download the push certificate from certificates section (.cer file)
Extract the .p12 part and add password (Can be done using openssl tool)
Copy the .p12 file to config\resources\mobile
Update the password as shown below –

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Table</th>
<th>PROP_ID</th>
<th>CATEGORY_ID</th>
<th>PROP_VALUE</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DIGX_FW_CONFIG_ALL_B</td>
<td>APNS</td>
<td>DispatchDetails</td>
<td>&lt;Password&gt;</td>
<td>Provides certificate password of the .p12 part</td>
</tr>
<tr>
<td>2</td>
<td>DIGX_FW_CONFIG_ALL_B</td>
<td>APNSKeyStore</td>
<td>DispatchDetails</td>
<td>DATABASE or CONNECTOR</td>
<td>Specifies whether to pick certificate password from database or from connector. Default DB (No change)</td>
</tr>
<tr>
<td>3</td>
<td>DIGX_FW_CONFIG_ALL_B</td>
<td>Proxy</td>
<td>DispatchDetails</td>
<td>&lt;protocol,proxy_address&gt;</td>
<td>Provides proxy address, if any, to be provided while connecting to APNS server. Delete row if proxy not required. Example: HTTP,148.50.60.8</td>
</tr>
<tr>
<td>4</td>
<td>DIGX_FW_CONFIG_ALL_B</td>
<td>CERT_TYPE</td>
<td>DispatchDetails</td>
<td>For dev push certs add row with value ‘dev’</td>
<td>For prod push certificates this row is not required</td>
</tr>
<tr>
<td>5</td>
<td>DIGX_FW_CONFIG_ALL_B</td>
<td>ios_cert_path</td>
<td>DispatchDetails</td>
<td>resources/mobile/ios-cert.p12</td>
<td>Update the certificate path/name if required. Should be relative to conf directory</td>
</tr>
</tbody>
</table>

If CONNECTOR is selected in Step 2 update password as below
Properties for tokens to be configured as –

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Table</th>
<th>PROP_ID</th>
<th>CATEGORY_ID</th>
<th>PROP_VALUE (Default Value)</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DIGX_FW_CONFIG_ALL_B</td>
<td>MOBILEJWT_EXPIRYTIME</td>
<td>dayoneconfig</td>
<td>864000</td>
<td>Time in secs after which user will have to reregister for alternate login in mobile app</td>
</tr>
</tbody>
</table>
3. **Archive and Export**

   a. In the Menu bar click on **Product -> Archive (Select Generic iOS Device)**

   b. After archiving has successfully completed. Following popup will appear

   c. Click on Export in the right pane of the popup -> **Distribute App -> Select method of distribution -> Choose Provisioning Profile -> select Export one app for all Compatible Devices -> Next -> Next and generate the ipa.**
4. **OBDX Authenticator Application**

4.1 **Building Authenticator UI**

1. Extract OBDX_Installer.zip. It contains OBDX_Installer/installables/mobile/authenticator/ui folder. The folder structure is as shown:

2. Open Terminal at “_build” level.

3. Run following command:

   ```bash
   sudo npm install -g grunt-cli
   sudo npm install
   node render-requirejs/render-requirejs.js
   grunt authenticator --verbose
   ```

4. After running above commands and getting result as “Done, without errors.” a new folder will be created at “_build” level with name as “dist”.

(b) **NON-OAM Based Authentication**

1. Copy “non-oam/login” folder and Replace it at location “components/modules/” [in ui folder] location. This will replace existing “login” folder.
2. Open Terminal at "_build" level.
3. Run following command:

```bash
sudo npm install -g grunt-cli
sudo npm install
node render-requirejs/render-requirejs.js
grunt authenticator --verbose
```
4. After running above commands and getting result as "Done, without errors." a new folder will be created at "_build" folder level with name as "dist".
4.2 Authenticator Application Workspace Setup

1. Unzip and navigate to iOS workspace as shipped in installer.
   a. Copy & Paste the cordova and CordovaLib as in section 2.2 Create Project above.

2. Open the “OBDX_Installer/installables/mobile/authenticator/ui/ios/www” folder in the finder and paste and replace the following generated UI files from “ui/dist” folder:
   - components
   - css
   - framework
   - images
   - pages
   - resources

Finally the Installer/installables/mobile/authenticator/ui/ios/www folder must look like:
3. Double click on **OBDXSecureAuthenticator.xcworkspace** to open the project in Xcode

Update HOTP or TOTP in above screenshots and update the server URL.

4. The application can be archived using steps in Section 4.3 for running on device

5. To run the application on simulator, copy & replace the framework from simulator/ObdxAuthenticator.framework to /authenticator/platforms/ios/
4.3 Building Authenticator Application

1. Set the simulator to Generic iOS device. Then go to Product -> Archive.

2. Choose your Archive. Click Distribute App as shown below and select appropriate profiles.

3. Click export to save the .ipa