Oracle® Banking Digital Experience Cloud Service

Mobile Application Builder Guide-iOS

Release 25.1.2.0.0
G51637-02
April 2026

Oracle Banking Digital Experience Cloud Service Mobile Application Builder Guide-iOS, Release 25.1.2.0.0

G51637-02

Contents

Index

Preface

Purpose

This guide is designed to help acquaint you with the Oracle Banking application. This guide provides answers to specific features and procedures that the user needs to be aware of the module to function successfully.

Audience

This document is intended for the following audience:

- Customers
- Partners

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=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also

mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Related Resources

For more information on any related features, refer to the following documents:

- Oracle Banking Digital Experience Cloud Service Licensing Manuals

Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes; actual screens that appear in the application may vary based on selected browser, theme, and mobile devices.

Acronyms and Abbreviations

The list of the acronyms and abbreviations used in this guide are as follows:

Table 1 Acronyms and Abbreviations

Abbreviation	Description
OBDXCS	Oracle Banking Digital Experience Cloud Service

1

OBDX Servicing Application on Saas

This topic provides information on **OBDX Servicing Application on Saas**. This is the main application for mobile banking.

- Prerequisite
  This topic provides information on **Prerequisite** to run IOS workspace.
- Project Setup
  This topic provides information on **Project Setup**.
- Create Project Using Remote UI
  This topic provides information on **Create Project Using Remote UI**.
- Configurations for the IOS application
  This topic provides information on **Configurations for the IOS application**.
- Enabling SSL pinning in the application
  This topic describes the systematic instruction to **Enabling SSL pinning in the application** option.
- Enabling Force update
  This topic provides information on **Enabling Force update**.
- Device Registration and Push Registration Functionality
  This topic provides information on **Device Registration and Push Registration Functionality**.
- Generating Certificates for Development, Production
  This topic describes the systematic instruction to **Generating Certificates for Development, Production** option.
- Setup for Push Notification in the application
  This topic describes the systematic instruction to **Setup for Push Notification in the application** option.
- Scan to Pay from Application Icon
  This topic describes the systematic instruction to **Scan to Pay from Application Icon** option.
- System Configuration on server for Application
  This topic provides information on **System Configuration on server for Application**.
- Changing App Icons and Assets
  This topic provides information on **Changing App Icons and Assets**.
- Archive and Export
  This topic describes the systematic instruction to **Archive and Export** option.

1.1 Prerequisite

This topic provides information on **Prerequisite** to run IOS workspace.

- Download and Install node as it is required to run npm and Cordova commands.
- Latest Xcode to be download from App Store. This document is w.r.t to Xcode 26.2.

- OBDX iOS Application support is provided for current iOS version and only one version preceding that.

1.2 Project Setup

This topic provides information on **Project Setup**.

Ensure **Nodejs Version is >= 12 and latest Xcode version available on AppStore is used.**

1. Extract iOS workspace from installer and place in a folder.
2. Mobile application is Cordova Hybrid Application.
3. The base core functionality is bundle in form of frameworks. The workspace contains fat xcframeworks for running on devices and simulator both. The same frameworks within the workspace can we used to run on simulator and device as well.
4. Below are the frameworks present inside the workspace. Verify if these are present before running the application on device or simulator.
    a. OBDXFramework.xcframework.
    b. CordovaFramework.xcframework.
    c. OBDXExtensions.xcframework.
    d. OBDXWatchFramework.xcframework.
5. Refer section: **Configurations for the IOS application** for configurations required for the application.
6. Apple certificates and provisioning profiles are needed to run the application on devices. Refer section: **Generating Certificates for Development Production**.
7. **Note**: OBDXCS is qualified on cloud for credential-based login, alternate login and snapshot functionality only.

1.3 Create Project Using Remote UI

This topic provides information on **Create Project Using Remote UI**.

UI will be deployed on Saas only, and the application will point to Saas URLs.

To enable UI debugging in network tab, set this property “InspectableWebView” to true in config.

1.4 Configurations for the IOS application

This topic provides information on **Configurations for the IOS application**.

Application-level configurations are present in ‘app.plist’ (ZigBank/Resources) of the workspace.

> **Note**
>
> These are configurations for different features. The description of each is in given below format.

Type - Data Type of value.

Purpose - Its usage.

Value – The possible values.

Configurable – Yes if bank can be allowed to change. No if the value is not allowed to be changed.

1. **For BACKEND URLs:**

   This is a mandatory configuration.

   OpenXcode by clicking ZigBank.xcodeproj at zigbank/platforms/ios/

   In your mobile application, set below properties as mentioned below:

   a. LOGIN_SCOPE: openid

   b. SERVER_TYPE: OAUTH3

   Replace below $ variables with actual values as per bank's setup: (Refer Mobile Application Cloud Configuration Guide.

   a. KEY_SERVER_URL: ${KEY_SERVER_URL}

   b. APP_CLIENT_ID: ${Client ID} (*as generated in IDCS*)

   c. IDCS_URL: ${IDSC_URL}/oauth2/v1/token

   d. WEB_URL: ${WEB_URL}

   e. REDIRECT_URI: ${REDIRECT_URI} (Set this value which is configured in IDCS mobile application)

   f. KEY_OAUTH_PROVIDER_URL: ${KEY_OAUTH_PROVIDER_URL}

2. **To Enable SSL:**

   a. By default, SSL pinning is NO in the workspace.

      Recommended to set to YES for production URLs with a valid authorized SSL certificate on server.

      For more information on fields, refer to the field description table.

Table 1-1 Enable SSL

SERVER_TYPE	Description
`CertificateType`	Type: String. Purpose: File extension of SSL Pinned certificates. Value: cer. Note: the certificate file added in the workspace should also have .cer extension. Configurable: No.
`PinnedUrl`	Type: Array. Purpose: Pinning URL to be entered here. This is the https URL of the server against which the certificate will be verified. Can add multiple if required. Value: https server URL. Configurable: Yes.

Table 1-1 (Cont.) Enable SSL

SERVER_TYPE	Description
PinnedCertificateName	Type: Array. Purpose: For verification of SSL, this certificate will be pinned in the application and verified against the server URL. Value: Houses the certificate name (without extension) of the pinning certificate. Old certificate (about to expire) and new one can co-exist. Configurable: Yes.
SSLPinningEnabled	Type: Boolean. Purpose: To enable SSL Pinning. SSL checks are performed on application launch. Value: YES for enabling. NO for disabling. Configurable: Yes.
SSLPinningEnabledNoNetworkCall	Provides the option of whether to load the login page if SSL Pinning fails. SSLPinningEnabled also must be set to YES for it to work. If set to YES and SSLPinningEnabled is set to YES then if SSL Pinning fails, then login page does not load. If set to NO and SSLPinningEnabled is set to YES then if SSL Pinning fails, then login page loads. Configurable: Yes.
EnableSSLPinningForEveryRequest	Type: Boolean. Purpose: To enable SSL Pinning for every request fired from application pages in the entire application. Value: YES, for enabling. NO for disabling. Configurable: Yes.

b. Refer section: **Enabling SSL pinning in the application** on how to configure the workspace to enable SSL pinning in the application.

3. **To Enable Force Update**

This is an optional configuration.

Refer section: **Enabling Force update** on more details on how to configure the workspace for this.

For more information on fields, refer to the field description table.

Table 1-2 Enable SSL

SERVER_TYPE	Description
ForceUpdate	Type: Boolean. Purpose: To enable force update feature in the application. Value: If set to YES, then the application will check for updates from the Appstore and display a non-dismissing popup. User needs to forcefully update the application. Default value: No. Recommended to set these configurations before releasing the first version so that force update works for future releases. Configurable: Yes.

Table 1-2 (Cont.) Enable SSL

SERVER_TYPE	Description
AppStoreID	Type: String. Purpose: The force update will be checked against this application ID. Value: Enter the ID of the application from AppStore. Configurable: Yes.
AppStoreURL	Type: String. Purpose: URL to AppStore redirection on click of update button. Value: It is set to https://itunes.apple.com/in/app/id@@AppStoreID?mt=8. Just replace @@AppStoreID to what is set above for 'AppStoreID'. Configurable: Just update as mentioned above. Do not change the URL.
itunesUrlForVersionCheck	Type: String Purpose: URL to check application version in AppStore for force update. Value: It is set to https://itunes.apple.com/in/app/id@@AppStoreID?mt=8. Just replace @@AppStoreID to what is set above for 'AppStoreID'. Configurable: Just update as mentioned above. Do not change the URL.

> **Note**
>
> If any of the above configurations are missing or invalid, the application will display blank screen on application launch.

4. **COMMON CONFIGURATIONS**

For more information on fields, refer to the field description table.

Table 1-3 COMMON CONFIGURATIONS

SERVER_TYPE	Description
XcodeBuildVersion	Build version with which the workspace is built with. Configurable: No.
PatchSetVersion	Version of the. OBDX application to identify the version of the workspace inside the patch installer.
SUITENAME	Group identifier for sharing keystore information. This Should match the app group added in the profile and in Targets->Signing Capabilities. App Groups are linked with the provisioning profile and its value can be verified from the Zigbank target->Signing Capabilities. This value is important for the secured storage of the information and SIRI to work. Configurable: Yes.
BankName	Name of bank to be shown on touch id / face id popup. Configurable: Yes.

Table 1-3 (Cont.) COMMON CONFIGURATIONS

SERVER_TYPE	Description
DomainDeployment	To be always set YES for token-based development. Configurable: No.

5. **For location tracking metrics**

   - This is optional. Bank needs to do if they need location tracking metrics for monitoring location-based data.

     For more information on fields, refer to the field description table.

Table 1-4 Location tracking metrics

SERVER_TYPE	Description
ALLOW_LOCATION_SHARE	By default, the value is false. If set to true, user will get location permission prompt to allow location tracking. It can be enabled if user's location needs to be tracked.

6. **For displaying "Rate Us" to redirect to Appstore page**

   - This is optional. User can have an option ("Rate Us") in settings to display App Store rating for the application. This option can be enabled/disabled from UI. Also, on click of the option, to open AppStore page for the application set below value. For more information on fields, refer to the field description table.

Table 1-5 Table Displaying "Rate Us" to redirect to Appstore page

SERVER_TYPE	Description
AppStoreURL	Replace @@AppStoreID with that of the application.

1.5 Enabling SSL pinning in the application

This topic describes the systematic instruction to **Enabling SSL pinning in the application** option.

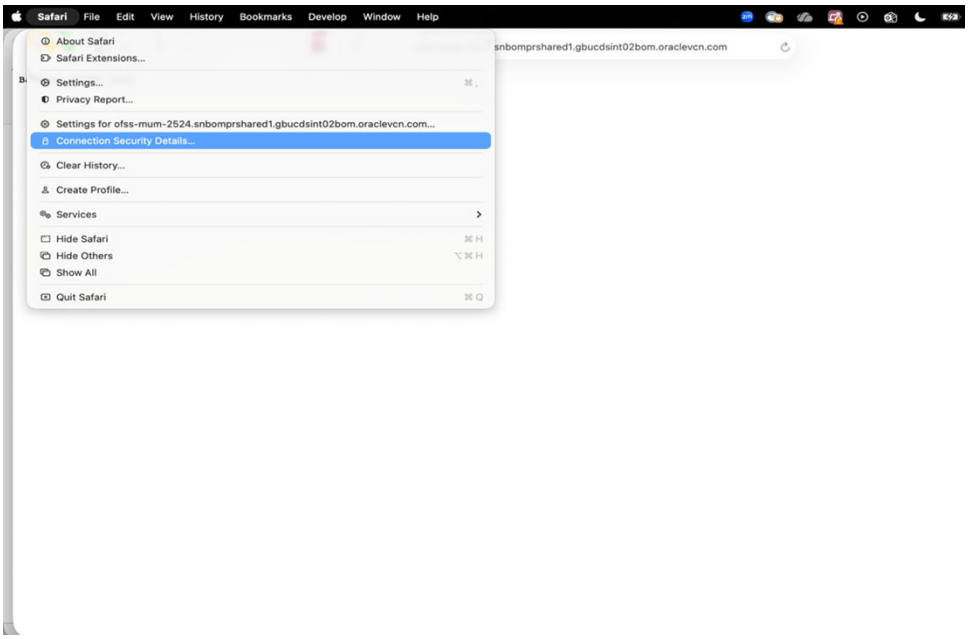
SSL pinning is developer using Apple Developer Guide.

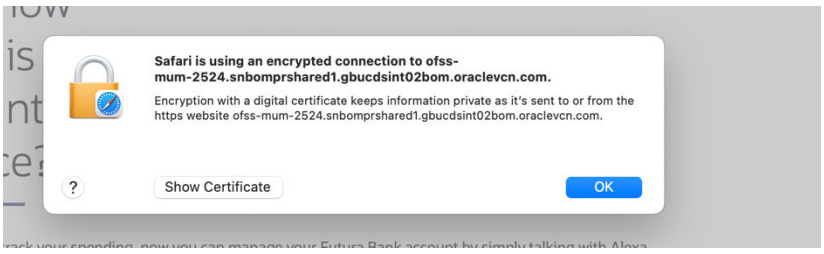
SSL pinning is required to securely connect with a https bank server URL to mitigate Man-in-middle-attack. It is recommended to enable this in production.

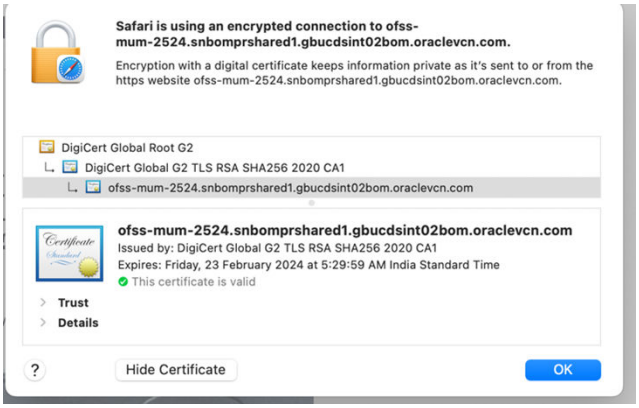
By default, SSL pinning is set to NO in the application for development purpose so that the application can connect to https URLs without SSL Pinning checks. Also, App Transport Security is disabled to allow http URLs.

> **Note**
>
> Full chain SSL certificate needs to be valid, and certificate should be issued from an authorized SSL authority. Self-signed certificate will be rejected by IOS OS itself. OS by default checks for a valid SSL trusted certificate using App Transport Security (ATS). Hence, the server should have a valid certificate chain and adhere to ATS requirements.
> Keep this App Transport Security (ATS) enabled for all the targets in IOS workspace for production-based application.

1. To enable SSL pinning, bank needs to follow the configurations mentioned in the section: **Configurations for the IOS application**.
2. The SSL certificate needs to be added in the workspace. To download and add this certificate, follow below steps:
   - Open bank's https website in Safari on Mac machine.
   - Go to Safari -> Connection Details.
   - Click on Show Certificate
   - Select the leaf certificate



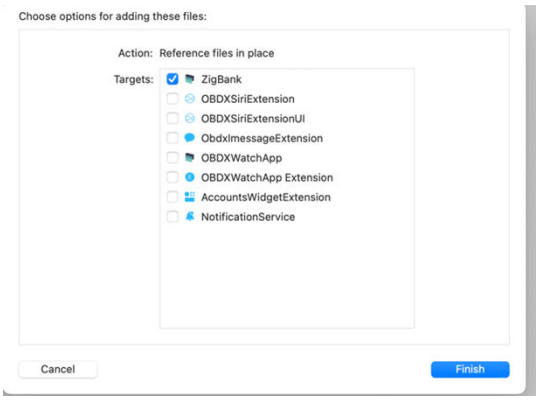




- Press and drag the certificate icon from Safari to any location on your machine.
- Rename it to any certificate.cer.
- Copy-paste the certificate inside IOS workspace at this location in

  /service/workspace_installer/zigbank/platforms/ios/ZigBank/

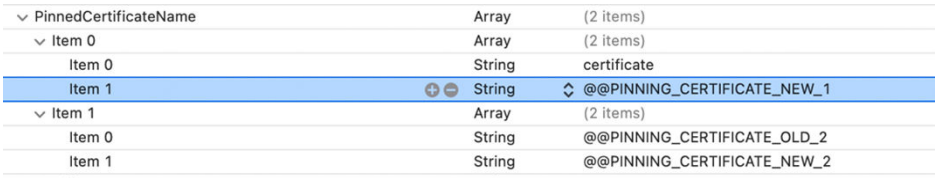
  .
- Right Click on Resources folder in Xcode and select “Add Files to Zigbank”.
- Select the certificate file which is saved in above step.
- Select ZigBank target.



- The certificate will be added in the Resources folder.
- Copy the name and add it in the app.plist against @@PINNING_CERTIFICATE_OLD_1 for PinnedCertificateName as shown below. Refer configuration section for this key information.

> **Note**
>
> Since this is an array, bank can add multiple certificates for @@PINNING_CERTIFICATE_OLD_1, @@PINNING_CERTIFICATE_OLD_2. Order doesn’t matter.

Also, since SSL certificate are renewed after the expiry @@PINNING_CERTIFICATE_NEW_1 and @@PINNING_CERTIFICATE_NEW_2 options are provided.



These are the corresponding new certificate names which can be added by the bank when the old certificates are about to expire and release this version of application to Appstore before the old certificate expires. This will allow that the application continues to work with SSL pinning even after old certificate has expired. Same activity bank can continue to do for every year before old certificate expires.

- To add the new certificates in workspace, bank must follow same steps as mentioned above.
- After the certificates are configured, next step is to set ‘PinnedUrl’ key in the app.plist. Refer configuration section for this key information. Add the https URL against which the certificates are to be verified. If there are multiple site certificates added, then bank must set all those URLs in each item as below:

1.6 Enabling Force update

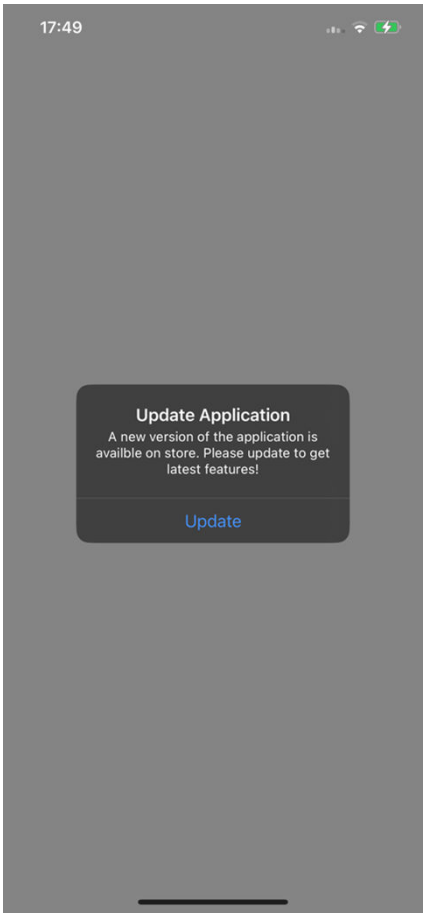
This topic provides information on **Enabling Force update**.

- This is an optional configuration but recommended.
- When a new version of the application is available on AppStore, user should be notified to upgrade their application. This flag will check the update and display a non-cancellable popup message to the user to update their application.
- For this to happen, enable this flag and other values as mentioned in the configuration section. Configurations for the IOS application.
- Set the version for the new update in “Bundle version string (short)” in ZigbankInfo.plist and in marketing version in Watch target->Build Settings. (Watch target setting is only required if Watch target is present for bank in its workspace.

> **Note**
>
> Bundle version string format should be same as that of the AppStore version set on iTunes developer portal. If Bank is uploading AppStore version 1.0 then the bundle version should be 1.0 and so on.

- So, if live app version is 1.0.1, the new application update can have version greater that 1.0.1 which can be either 2.0 or 1.0.2 or 1.1.0 or 1.1.1 etc. Once this new version is released to Appstore, the application already installed on user’s device will compare the installed version with new updated version.

- If updated version is available, then below popup will be displayed.



On clicking the button, user will be redirected to the Appstore page of that application. (Ensure to set correct AppStoreID in the configuration for this redirection)

- The popup header text and message can be configured in “Localizable.strings” file inside Classes folder in the workspace for below keys:

```
Header – APP_UPDATE_HEADER
Message - APP_UPDATE_TEXT
Button text- APP_UPDATE_BTN_TEXT
```

- This feature will work only after from the time the users install the version which has this logic enabled. Ex: If bank has 1.0 in Appstore for which this flag as false and bank releases 1.1 in Appstore with this flag enabled, then user needs to install the 1.1. application manually. Since 1.0 didn’t have that flag enabled, it will not check for any updates. However, every future release made to Appstore will check for any updates and display the force update popup.

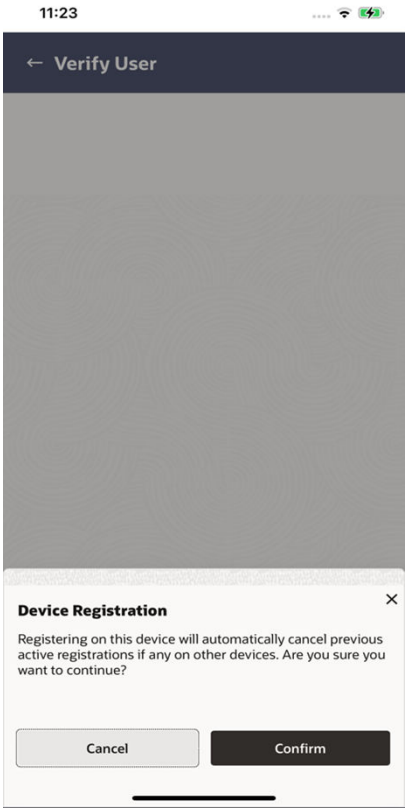
1.7 Device Registration and Push Registration Functionality

This topic provides information on **Device Registration and Push Registration Functionality**.

1. Device registration is used for alternate login registrations. Apple does not allow any unique identifier or device UUID to be used. Device is identified by a unique ID generated

on application life cycle when the application is installed. New device Id will be generated after is re-installed or alternate logic is de-registered.

2. Push registration is used for registering Push tokens for delivering push. These tokens are generated by Apple and registered in OBDX server. These tokens are specific to development and production environment. The configurations are mentioned in the system configuration section.
3. Bank can allow single or multiple devices to be registered for push and alternate login.
4. Consider a case when a single device is allowed to be registered for alternate login, then only one device registration will be active. If user tries to register another device with same username for alternate login, then the previous registration of other older devices will be removed.
5. While user registers his second device or same device again (by re-installing the application), a popup will appear to notify the same.
6. If user confirms, then the current device will be registered, and all previous registrations will be removed.



If user cancel, the process is exited.

7. User will get an error message if he/she tries to login using PIN/PATTERN/FACE on the de-registered or old devices.
8. Device registration count is controlled by ALLOWED_DEVICE_COUNT to any value between than 1 and 100.
   - 1 will allow on one device registration.
   - 100 will allow more than one device registration.
   - Refer section: System Configuration on server for more details.

9. Push registration count is controlled by ALLOWED_PUSH_DEVICE_COUNT any value between 1 and -1.
    - 1 will only one device to be registered for push.
    - -1 will only multiple devices to be registered for push.
    - Refer section: System Configuration on server for more details.

1.8 Generating Certificates for Development, Production

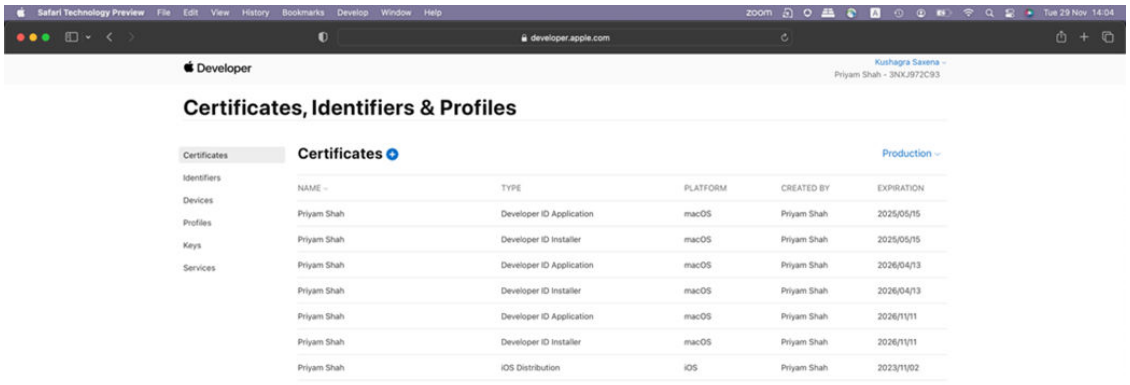
This topic describes the systematic instruction to **Generating Certificates for Development, Production** option.

1. This is required for running the application on device for debugging, testing as well for releasing the application to Appstore.
2. Bank can refer to Apple's documentation on how to create certificates and provisioning profiles.
3. Create all certificates (by uploading CSR from keychain utility), provisioning profiles and push certificates by login in developer console.

Below are steps:

1. Certificate Creation.
2. AppID creation.
3. Profile creation.
4. Adding device UUIDS to profiles.
5. Generating Push certificate for server.

Certificate Creation: Below is the screen on apple developer portal where bank needs to create distribution and Development certificates.



AppID creation: Below is the screen where bank needs to create appIDs for each target bank has configured in workspace. Available targets are:

ZigBank

OBDXSiriExtension

OBDXSiriExtensionUI

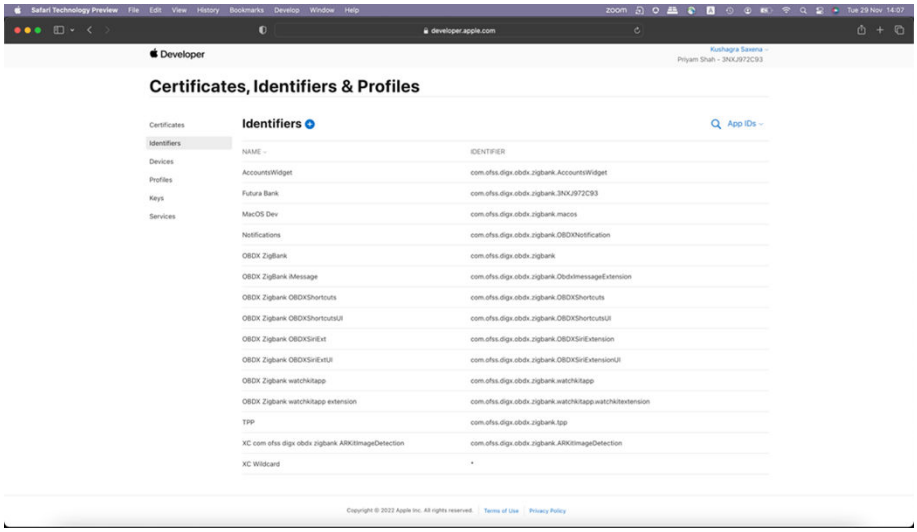
ObdxImessageExtension

OBDXWatchApp

OBDXWatchApp Extension

AccountsWidgetExtension

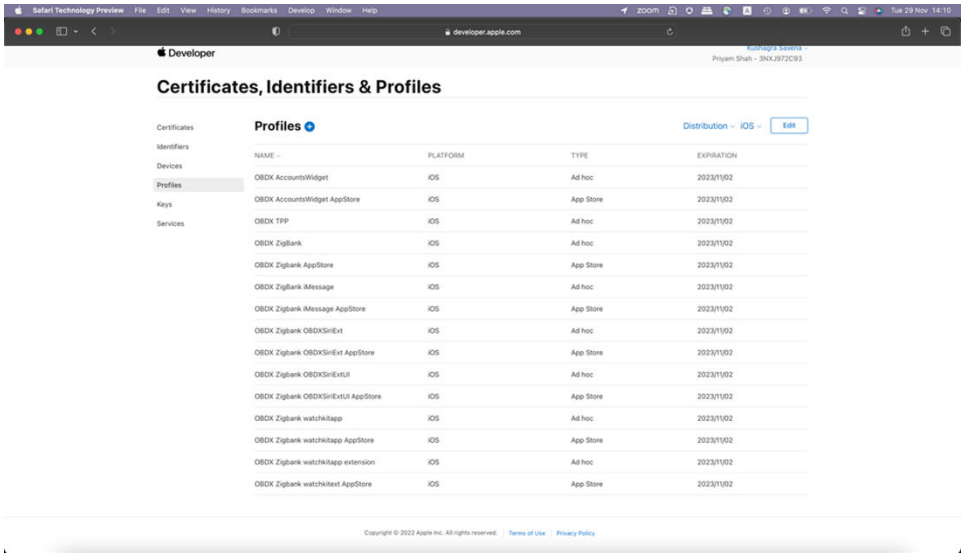
NotificationService



1. Add capabilities as shown below and ensure the bundle identifier matches with the capabilities added in Xcode.
2. Ensure AppGroups capability is added to all profiles and for appIDs.
3. Ensure SiriKit, App Groups, Push Notifications, Associated domain capabilities are added in Zigbank appIDs.
4. Bank can refer base workspace for the naming convention followed for bundle identifier for each target. Below are the appIDs which we need for OBDX application in similar format as below:

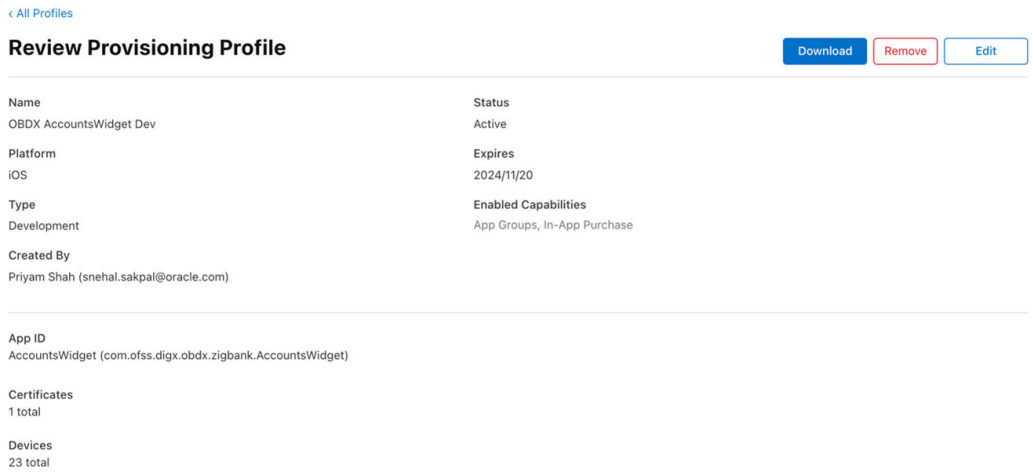
OBDX ZigBank iMessage	com.ofss.digx.obdx.zigbank.ObdxImessageExtension
OBDX Zigbank OBDXSiriExt	com.ofss.digx.obdx.zigbank.OBDXSiriExtension
OBDX Zigbank OBDXSiriExtUI	com.ofss.digx.obdx.zigbank.OBDXSiriExtensionUI
OBDX Zigbank watchkitapp	com.ofss.digx.obdx.zigbank.watchkitapp
OBDX Zigbank watchkitapp extension	com.ofss.digx.obdx.zigbank.watchkitapp.watchkitextension
OBDX Notification Extension	com.ofss.digx.obdx.zigbank.OBDXNotificationExtension
OBDX ZigBank	com.ofss.digx.obdx.zigbank
AccountsWidget	com.ofss.digx.obdx.zigbank.AccountsWidget

Profile Creation:



Select appropriate AppIDs to relevant profile and appropriate certificates.

Example: AccountWidget development profile will have development certificate and appId created for AccountWidget. Likewise for other targets.



Bundle identifiers need to be added in the Info.plist of each frameworks. Example: if bank's appId for Zigbank is com.ofss.digx.obdx.zigbank then follow below steps:

1. Right click on OBDXFramework.xcframework(in Xcode's Project Navigator) -> Show in Finder.
2. When the finder directory opens the right click OBDXFramework.xcframework -> select ios-arm64 -> Select OBDXFramework.framework.
3. Open Info.plist and set Bundle identifier as the one created for bank's application. Example: com.ofss.digx.obdx.zigbank.OBDXFramework.
4. Repeat the steps for the other three frameworks as well, with the following values: Bundle identifier for Cordova.framework : com.ofss.digx.obdx.zigbank.Cordova.

   Bundle identifier for OBDXExtensions.framework : com.ofss.digx.obdx.zigbank.OBDXExtensions.

   Bundle identifier for OBDXWatchFramework.framework : com.ofss.digx.obdx.zigbank.OBDXWatchFramework.

Set the identifier in the Signing Capabilities tab in Xcode for each target.

1. Open Xcode project in Xcode and select each target and go to Signing and Capabilities and update correct bundle identifier for each target.
2. Example if main target bundle identifier is "com.ofss.digx.obdx.zigbank" then each target should have below format for bundler identifiers:
   OBDXSiriExtensionSiri – com.ofss.digx.obdx.zigbank.OBDXSiriExtension

   OBDXSiriExtensionUI – com.ofss.digx.obdx.zigbank.OBDXSiriExtensionUI

   ObdxImessageExtension– com.ofss.digx.obdx.zigbank.ObdxImessageExtension

   OBDXWatchApp – com.ofss.digx.obdx.zigbank.watchkitapp

   OBDXWatchAppExtension-com.ofss.digx.obdx.zigbank.watchkitapp.watchkitextension

   AccountsWidgetExtension – com.ofss.digx.obdx.zigbank.AccountsWidget

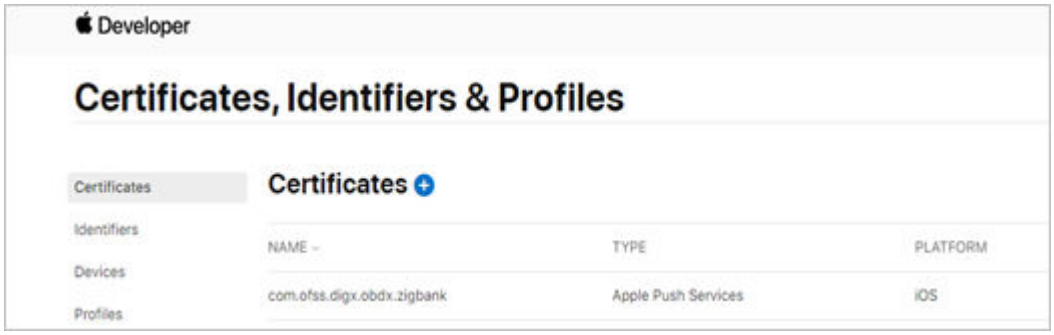
   NotificationExtension- om.ofss.digx.obdx.zigbank.OBDXNotificationExtension

Adding device UUIDS to profiles

1. For development profiles, testing device UUIDs need to be added, and same devices need to be selected in the development profile.

Generating Push certificate for server

To set up an APNs certificate on your server, bank will need to generate a Certificate Signing Request (CSR), upload it to Apple's Push Certificates Portal, download the resulting certificate, and then install it on your server, along with the necessary root certificates for secure communication.

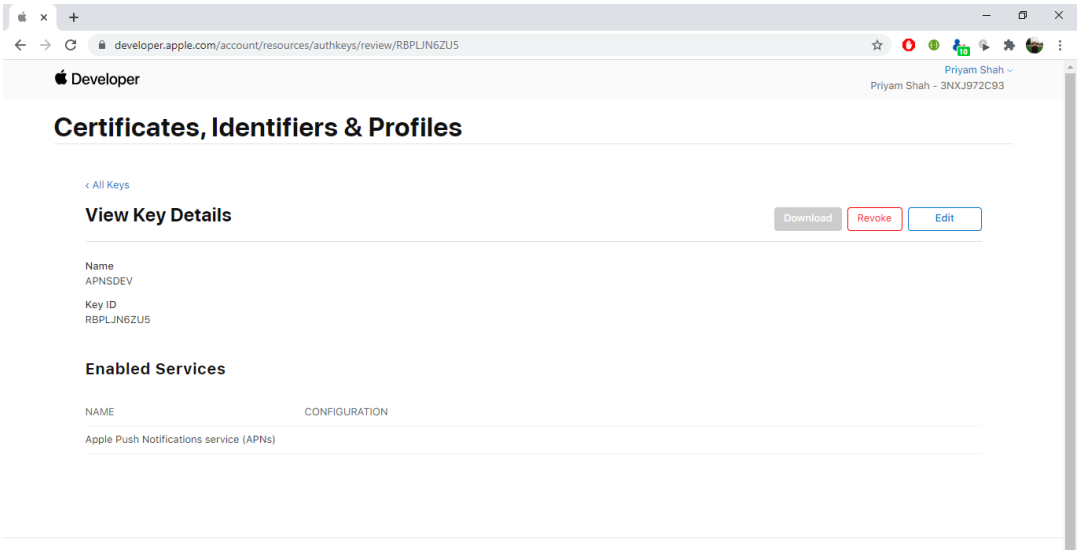


1.9 Setup for Push Notification in the application

This topic describes the systematic instruction to **Setup for Push Notification in the application** option.

1. Push notification services are to be created using p8.
2. For p8, bank needs to setup Key and update database with the details. All details are mentioned below:
3. Create APNS key from developer portal. Navigate to the “Keys” section and create APNS key.

> **Note**
>
> APNS key and download the .p8 file. We need the contents of this file to update in database.

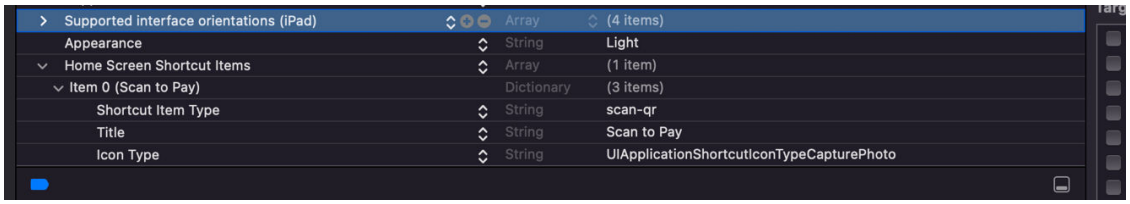


1.10 Scan to Pay from Application Icon

This topic describes the systematic instruction to **Scan to Pay from Application Icon** option.

Users can long press on bank's application icon on home screen and click on scan-to-pay option to scan QR and make payments.

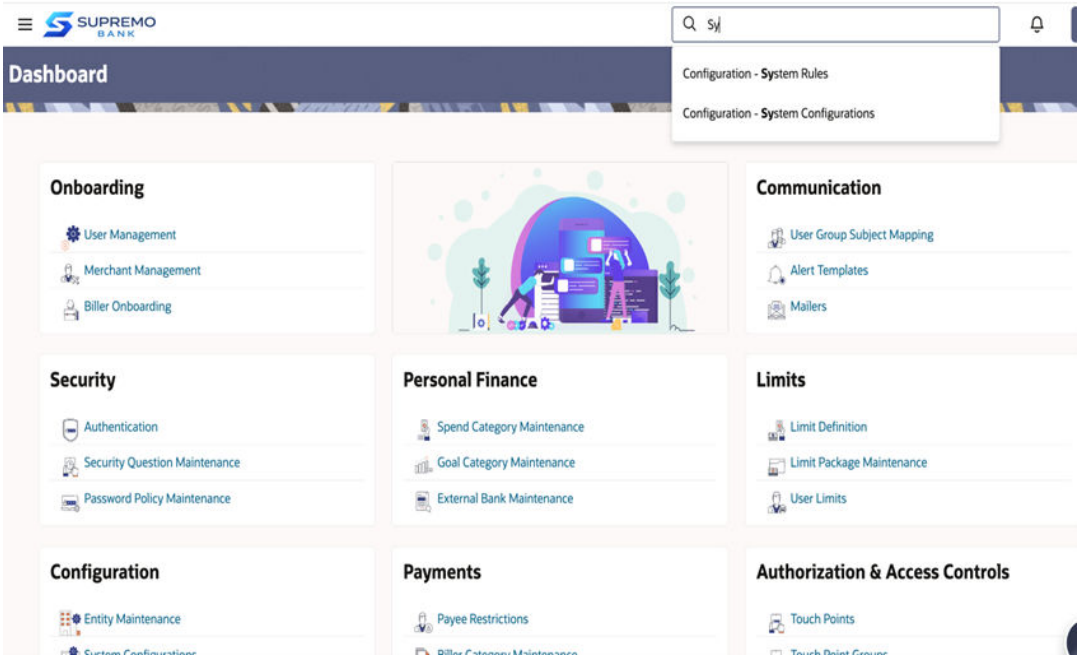
1. This option is not RTM controlled hence to remove this option if bank doesn't need it, then open Zigbank project in Xcode, open ZigBank-Info.plist. Delete entry for key – "Home Screen Shortcut Items".



1.11 System Configuration on server for Application

This topic provides information on **System Configuration on server for Application**.

1. Proxy password to be mounted by the cloud AMS team
2. Update below configurations for IOS Push notifications from OBDCS web console using super admin login. Login to OBDCS using admin and Open System Configurations Screen.



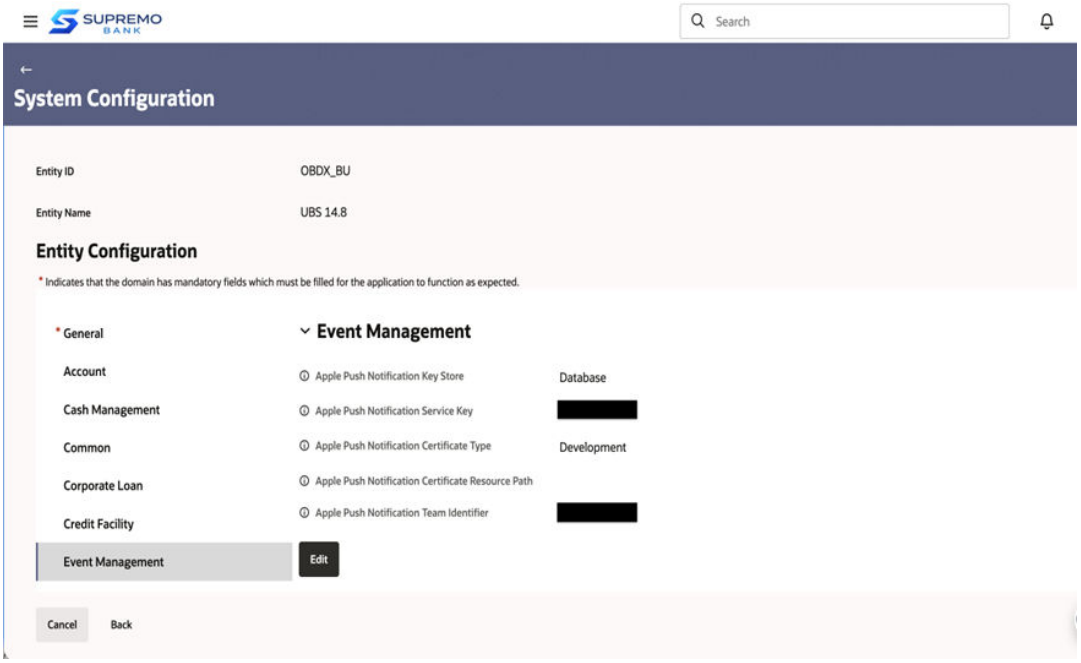


Table 1-6 Configurations for IOS Push notifications

Name	Path for the config	Value
Apple Push Notification Key Store	Entity -> Event Management	Database. (Do not change this)
Apple Push Notification Service Key	Entity -> Event Management	<APNS key generated on apple Developer console>

Table 1-6 (Cont.) Configurations for IOS Push notifications

Name	Path for the config	Value
Apple Push Notification Certificate Type	Entity -> Event Management	Set it to Development for dev-based key. Set it to Production for prod-based key.
Apple Push Notification Team Identifier	Entity -> Event Management	Bank's Team Identifier on Apple Developer Console
Apple Push Notification Key Certificate Store	Entity -> Event Management	Database. (Do not change this)
Apple Push Notification Key Certificate	Entity -> Event Management	Based 64 of the APNS key certificate generated from Apple Developer Console.
Apple Push Notification Bundle Identifier	Entity -> Event Management	Bank's Bundle Identifier for the application on Apple Developer. This is same as set for Zigbank target in the IOS workspace.
Proxy settings	Entity -> Event Management	Proxy settings to allow APNS connection from OBDX server. The format is: HTTP,<proxy>,<proxy port>
Allowed Push Registration Count	Global Settings ->General	In base, it is configured as 1. Set to 1 or more than 1 according to bank's requirement to allow those many devices for push per user.

3. Biometric Settings

Table 1-7 Biometric Settings

Name	Path for the config	Value
Allowed Device Count for Biometric Registration	Global Settings -> Framework	Set to 1 or 100 based on bank's requirement to allow number of devices or biometric registration

1.12 Changing App Icons and Assets

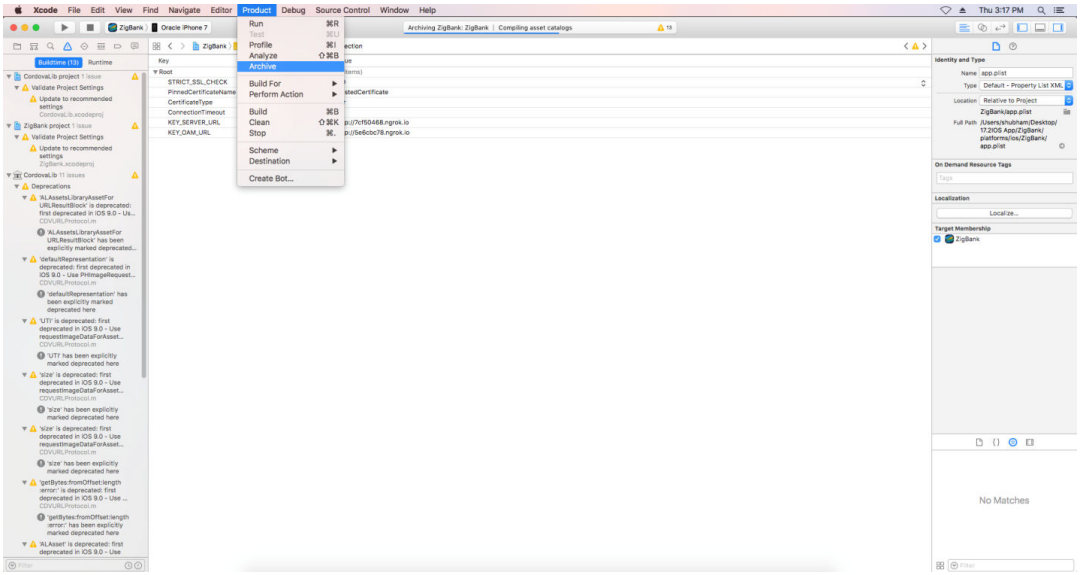
This topic provides information on **Changing App Icons and Assets**.

1. All the app icons for all sizes and AppStore icon needs to be replaced in Images.xcassets file under Zigbank → Resources folder. Refer image properties from Base Xcode added images for alpha channel etc. Also, detailed level sizes can be obtained from Apple Documentation.
2. The launch images to be added in Images.xcassets file under Zigbank → Resources folder.
3. The application as well shows splash images when application goes in background. These images are present in Zigbank → Classes folder with names prefixed "Default-". Replace all 4 images with appropriate size. Use high resolution image sizes 1125x2436 for crisp images on high resolution devices.

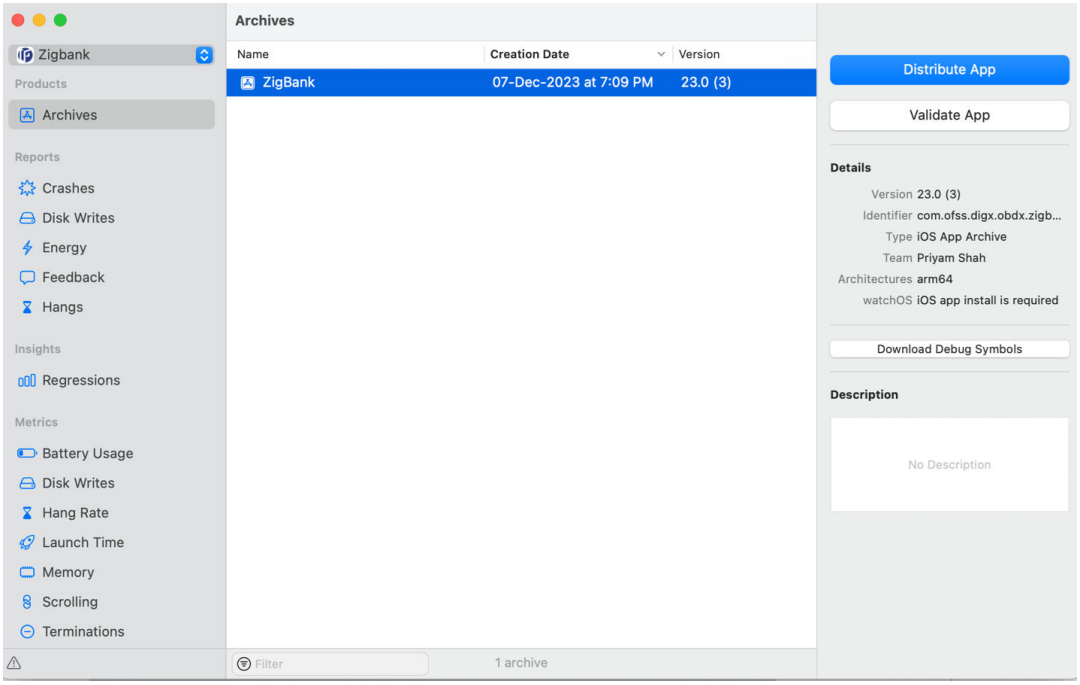
1.13 Archive and Export

This topic describes the systematic instruction to **Archive and Export** option.

1. In the menu bar, click on **Product → Archive (Select Generic iOS Device).**



2. After archiving has successfully completed. Following popup will appears.



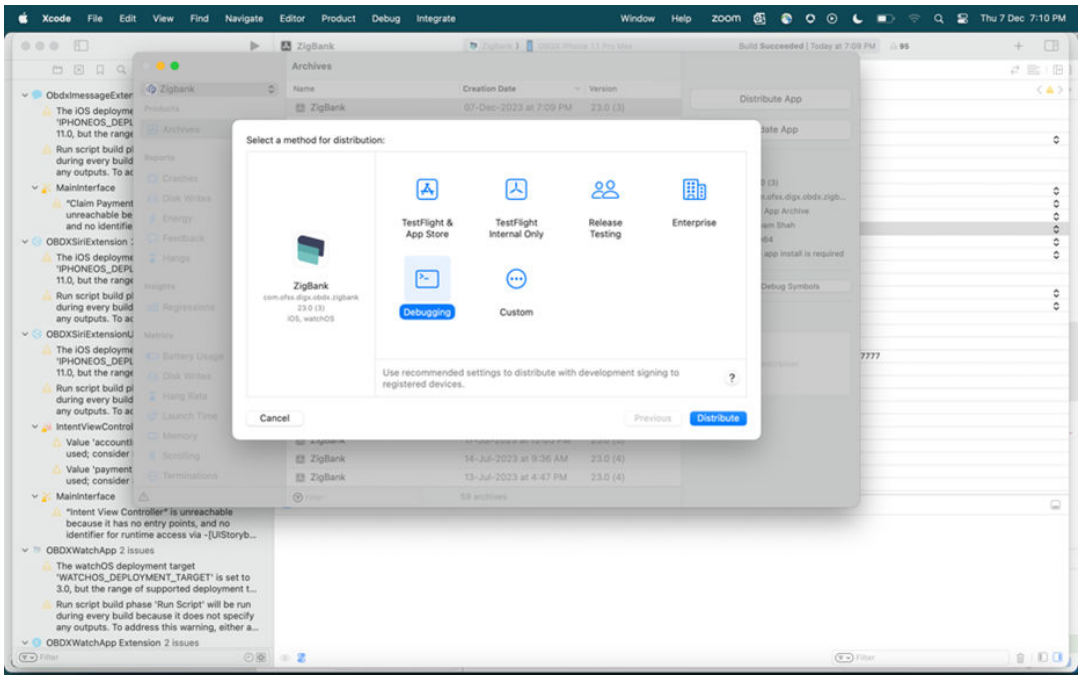
3. Click on Distribute App in the right pane of the popup → select **the Method of Distribution → Select Distribute.** Review the contents and click on **Export → Export** and generate the. ipa

   a. There are multiple options for exporting, select according to what is needed.

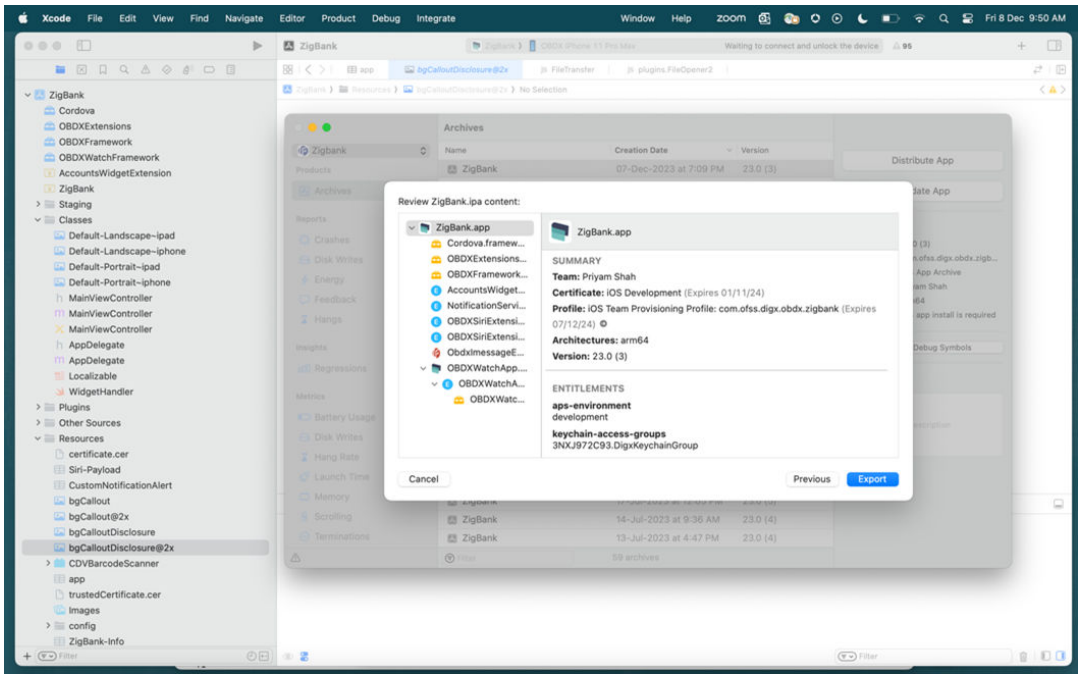
b. Debugging – this will create an ipa with development profile for internal testing.

c. Release Testing – This will create an ipa with Ad Hoc distribution profile for adHoc testing.

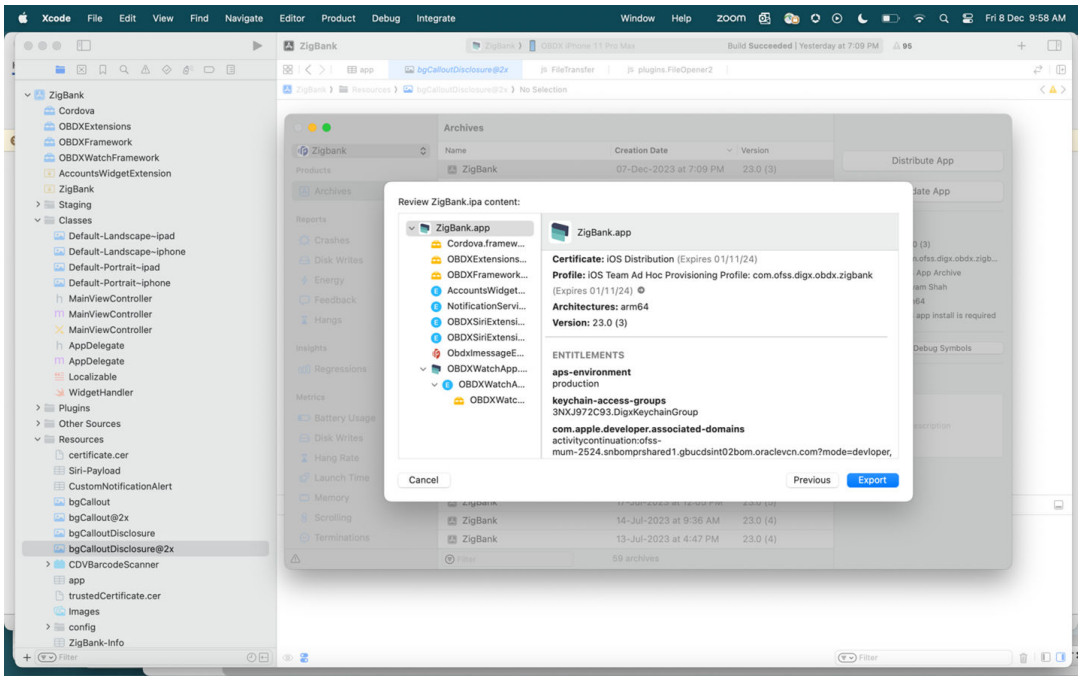
d. TestFlight Internal Only, TestFlight & AppStore- As the name suggests, this is for TestFlight and AppStore release.

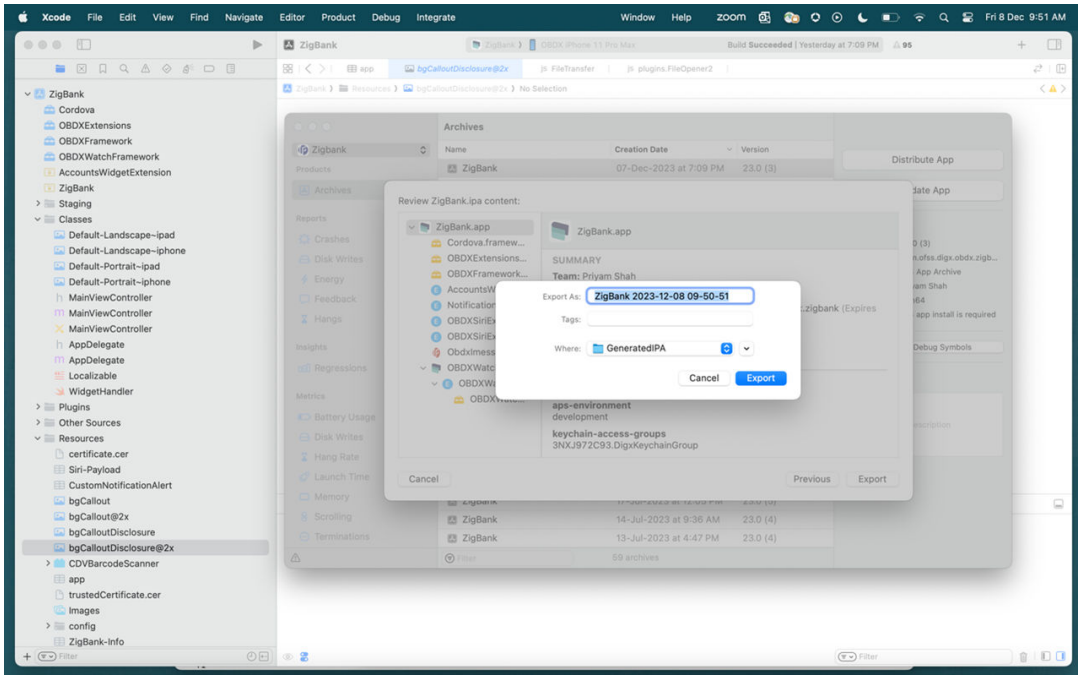


This is window which appear after selecting Debugging option. Note the Certificate and provisioning profile is for development type ipa likewise for other types verify the same.



4. Click on Export and it will ask to save the ipa. Select the location and click on Export. This ipa will be development ipa which can be installed on devices which are added in the profiles on developer account.
Below is the window which appears after selecting “Release Testing”. Note here the Certificate and Profile is of Adhoc Distribution.





Follow the above steps to Export and save the ipa. This ipa will be adhoc distribution ipa.

5. The application can be pushed to TestFlight for test flight testing. Refer Apple Documentation for Test Flight Set up.

2

Apply Privacy

This topic describes the systematic instruction to **Apply Privacy** option.

Apple requirements for Required Reason's Api and Data Types usage

> **Note**
>
> This document is for bank's reference for the Apple's rejection issue related to "Required Reason's Api and Data Types usage".

References:

WWDC 2023 video.

https://developer.apple.com/support/third-party-SDK-requirements/

https://developer.apple.com/documentation/bundleresources/privacy_manifest_files/adding_a_privacy_manifest_to_your_app_or_third-party_sdk#4336740

What's needs to be done in the application

1. Since apple has pointed out Cordova as thirds party SDK which needs to add this, we have added "PrivacyInfo.xcprivacy" inside cordova framework and added below items for required reason Api and data types. Bank should use the latest Cordova.xcframework.

   > **Note**
   >
   > We have added as per what we have used inside cordova and Apple's documentation, however Apple's Email with details will be more useful to target the required keys to be added.

```
<?xml
            version="1.0" encoding="UTF-8"?><!DOCTYPE plist PUBLIC "-//
Apple//DTD PLIST 1.0//EN"
            "http://www.apple.com/DTDs/PropertyList-1.0.dtd"><plist
version="1.0"><dict>
            <key>NSPrivacyAccessedAPITypes</key>
          <array>
            <dict>

            <key>NSPrivacyAccessedAPIType</key>

            <string>NSPrivacyAccessedAPICategoryDiskSpace</
string>

            <key>NSPrivacyAccessedAPITypeReasons</
key>
            <array>
```

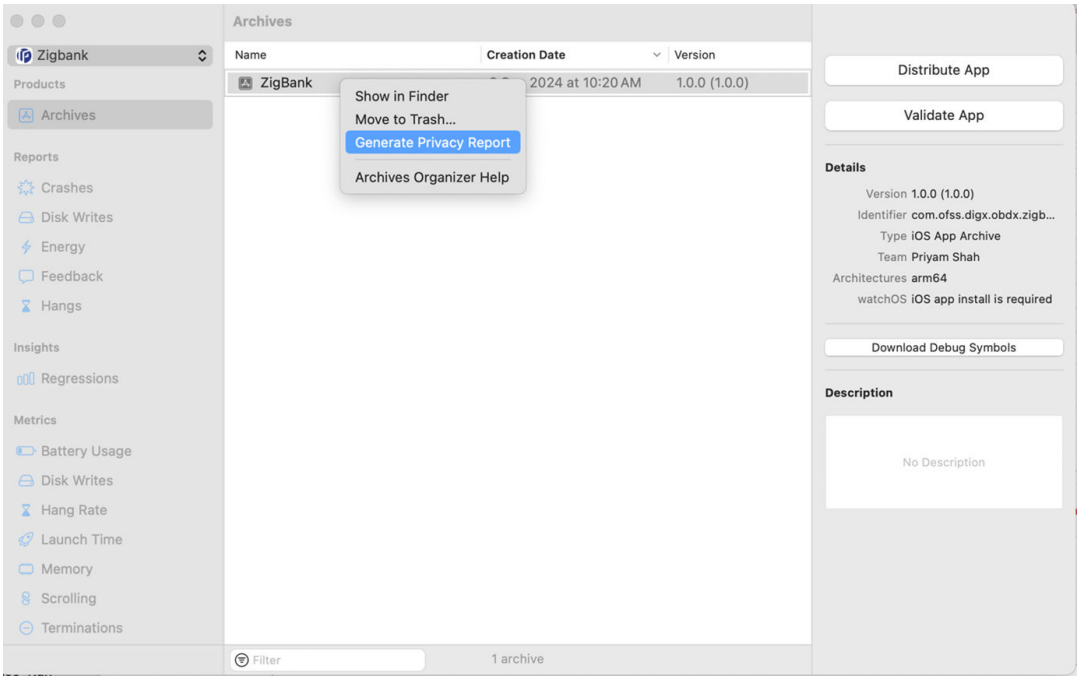
```
                <string>E174.1</string>
                </array>
                </dict>
                <dict>

                <key>NSPrivacyAccessedAPITypeReasons</
key>
                <array>
                <string>C617.1</string>
                </array>

                <key>NSPrivacyAccessedAPIType</key>
                   <string>NSPrivacyAccessedAPICategoryFileTimestamp</
string>
                </dict>
                <dict>

                <key>NSPrivacyAccessedAPITypeReasons</
key>
                <array>
                <string>1C8F.1</string>
                </array>

                <key>NSPrivacyAccessedAPIType</key>
                   <string>NSPrivacyAccessedAPICategoryUserDefaults</
string>
                </dict>
              </array>
                <key>NSPrivacyTracking</key>
              <false/>
                <key>NSPrivacyCollectedDataTypes</key>
              <array>
                <dict>
            <key>NSPrivacyCollectedDataType</key>

                <string>NSPrivacyCollectedDataTypeDeviceID</
string>

                <key>NSPrivacyCollectedDataTypeLinked</
key>
                <true/>

                <key>NSPrivacyCollectedDataTypeTracking</
key>
                <false/>

                <key>NSPrivacyCollectedDataTypePurposes</
key>
                <array>
                <string>App functionality</
string>
                </array>
                </dict>
              </array></dict></plist>
```

2. Additionally, there is “PrivacyInfo.xcprivacy” file added at Zigbank target level inside Resources folder.

   Bank needs to add additional items as per their bank customize code in “PrivacyInfo.xcprivacy” file. They can refer Apple documentation link for further details. Also, Apple’s mail can contain details of what all is missing in the PrivacyInfo.xcprivacy. Those items can be added as per Apple’s doc.

   > **Note**
   >
   > This step is not mandatory but if there any reference of such file in Apple’s mail, bank needs to add privacy items details in application target’s “PrivacyInfo.xcprivacy” file.

3. Once added, build can be archived, and in organizer, right click on the application and Generate Privacy Report. This report will have the only the details of Nutrition label added in the application. Check if all nutrition labels declared in “PrivacyInfo.xcprivacy” file are present in the generated reported.



4. Generate the application and upload to Appstore for Apple Review.

3

Make IOS Application Ready for Production Checklist

This topic provides information on **Make IOS Service Application Ready for Production**.

Apart from Apple AppStore Submission guidelines, below are checklists in IOS workspace:

- Confirm that the bundle identifiers and profiles are AppStore Profiles.
- Confirm the suit name in app.plist is matching App Groups mapped to profile and set in Target settings.
- Server URL is correct in app.plist. Recommended is to use https URL with SSL certificate from a valid trust Authority.
- Recommended to enable SSL Pinning. If enabled, check all the configurations as stated in configuration section.
- App icons, splash images are updated.
- Bundle version is set in “Bundle version string (short)” in info.plist and in ZigBank Target → Info and in marketing version in Watch target → Build Settings.
- Push notification configurations are set to prod in config table and APNS keys are correct.
- If Associated domains are enabled for usage, then remove "?mode=developer" from “Associated Domains” in Zigbank target.
- Other IOS workspace configurations are followed as per section Configurations for the IOS application.
- App Transport security is enabled by setting “Allow Arbitrary Loads” to NO for all target’s info.plist. Test this once with production URL.
- InspectableWebView property is set to false in config.xml.

Index