

Oracle® Retail Store Operations Cloud Service

Mobile Guide



Release 25.0.101.0

G18418-03

April 2025

ORACLE®

Oracle Retail Store Operations Cloud Service Mobile Guide, Release 25.0.101.0

G18418-03

Copyright © 2025, Oracle and/or its affiliates.

Primary Author: Anja Schafstall

Contributing Authors: Brian Voelz, Bipin Pradhan

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Send Us Your Comments

Preface

Audience	vi
Documentation Accessibility	vi
Related Documents	vi
Improved Process for Oracle Retail Documentation Corrections	vii
Oracle Retail Documentation at the Oracle Help Center	vii
Conventions	vii

1 Preinstallation Details

Technical Concepts	1-1
Technical Specifications	1-1
MAF 2.6.6 Runtime Support and Certification Matrix	1-1

2 Installing and Setting up the Application — MAF Mobile

Store Operations Cloud Service	2-1
Prerequisites	2-1
Platform Specific Requirements	2-2
Installer Artifacts	2-2
Installing the MAF Extension	2-3
Configuring JDeveloper for iOS Deployment	2-3
Configuring JDeveloper for Android Deployment	2-4
Creating a Workspace from the Archive	2-5
Configuring Application Settings	2-7
Definitions of SOCS Configuration Service End Points	2-7
Configuration Service Setup	2-9
No Configuration Service Setup – Application Ready for Use	2-10
Navigation Configuration	2-10
Customizing Application Branding	2-12
Customizing Application Skins	2-12

Customizing String Resources	2-13
Resource Bundles	2-14
Configuring Application for iOS Deployment	2-14
Deploying Application to Distribution Package to Generate .ipa File	2-14
Configuring Application for Android Deployment	2-16
Deploying Application to Distribution Package to Generate .apk File	2-17
Android Scanning Support	2-18
Configuring DataWedge Profile for Scanning	2-18
Details on Item Images	2-20
Image Sizing	2-20
Thumbnail Images	2-20
High Resolution Images	2-21
Configurable Threads	2-21

3 Installing and Setting up the Application - JET Mobile

Prerequisites	3-1
Platform-Specific Requirements	3-1
Pre-Installation Notes	3-1
Configuring Application Settings	3-2
Definitions of JET Mobile	3-2
Configuring DataWedge Profile for Scanning	3-3
Configuring Oracle MICROS devices for Scanning	3-3
Building JET Mobile as an iOS Application	3-4
Prerequisites	3-4
Building the Application	3-4
Contents of main.m	3-5
Running JET Mobile as a Browser-Based Application	3-5
Details on Item Images	3-6
Image Sizing	3-6
Thumbnail Images	3-6
High Resolution Images	3-6
Network Considerations	3-6

Send Us Your Comments

Oracle Retail Store Operations Cloud Service Mobile Guide

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).



Note:

Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Help Center (OHC) website. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at <http://www.oracle.com>.

Preface

This guide describes the requirements and procedures to install this Oracle Retail Store Operations Cloud Service (SOCS) Mobile release.

Audience

This Guide is for the following audiences:

- System administrators and operations personnel
- Database administrators
- System analysts and programmers
- Integrators and implementation staff personnel

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://docs.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 <https://docs.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <https://docs.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Retail Store Operations Cloud Service documentation set:

- *Oracle Retail Store Inventory Operations Cloud Services Release Notes*
- *Oracle Retail Store Inventory Operations Cloud Services Implementation Guide*
- *Oracle Retail Store Inventory Operations Cloud Services Data Model*
- *Oracle Retail Enterprise Inventory Cloud Service Administration Guide*
- *Oracle Retail Enterprise Inventory Cloud Service Inbound and Outbound Integration Guide*
- *Oracle Retail Enterprise Inventory Cloud Service Security Guide*
- *Oracle Retail Enterprise Inventory Cloud Service User Guide*
- *Oracle Retail Store Operations Cloud Service User Guide*

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced at the Oracle Help Center (OHC) website, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available at the Oracle Help Center at the following URL:

<https://docs.oracle.com/en/industries/retail/index.html>

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number F123456-02 is an updated version of a document with part number F123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

Oracle Retail Documentation at the Oracle Help Center

Oracle Retail product documentation is available on the following website:

<https://docs.oracle.com/en/industries/retail/index.html>

(Data Model documents are not available through Oracle Help Center. You can obtain them through My Oracle Support.)

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.
<i>italic</i>	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.

1

Preinstallation Details

Technical Concepts

The installer should understand the following technical concepts:

- JDeveloper
- Application servers
- XML manipulation
- Apple Enterprise Development setup, provisioning, and deployment
- Xcode
- Android SDK Tools, Platform-tools, Build-tools and Deployment
- Certificates management (generating keystores, import/exporting certs)

Technical Specifications

MAF 2.6.6 Runtime Support and Certification Matrix

Table 1-1 Runtime Support and Certification Matrix

Mobile Operating System	Version	MAF Supported Devices
Android	11.x	Any device running Android 11.0 or above with Minimum 1 GHz processor / 1 GHz Dual Core or better (Recommended) and at least 1 GB of total RAM.
Android	10.x	Any device running Android 10.0 or above with Minimum 1 GHz processor / 1 GHz Dual Core or better (Recommended) and at least 1 GB of total RAM.
Android	9.x	Any device running Android 9.0 or above with Minimum 1 GHz processor / 1 GHz Dual Core or better (Recommended) and at least 1 GB of total RAM.
Android	8.x	Any device running Android 8.0 or above with Minimum 1 GHz processor / 1 GHz Dual Core or better (Recommended) and at least 1 GB of total RAM.
Android	7.x	Any device running Android 7.0 or above with Minimum 1 GHz processor / 1 GHz Dual Core or better (Recommended) and at least 1 GB of total RAM.
Android	6.x	Any device running Android 6.0 or above with Minimum 1 GHz processor / 1 GHz Dual Core or better (Recommended) and at least 1 GB of total RAM.

Table 1-1 (Cont.) Runtime Support and Certification Matrix

Mobile Operating System	Version	MAF Supported Devices
iOS	10.15 (Catalina)	All Devices
iOS	10.14.4 (Mojave)	All Devices

For more information on build time support, see the Oracle Mobile Application Framework Certification Matrix at:

<https://www.oracle.com/application-development/technologies/maf/v266certification-matrix.html>

2

Installing and Setting up the Application — MAF Mobile

This chapter describes how to install and configure JDeveloper and Oracle Store Operations Cloud Service (SOCS) for deployment onto an iOS or Android device.

- [Prerequisites](#)
- [Installer Artifacts](#)
- [Installing the MAF Extension](#)
- [Configuring JDeveloper for iOS Deployment](#)
- [Configuring JDeveloper for Android Deployment](#)
- [Creating a Workspace from the Archive](#)
- [Configuring Application Settings](#)
- [Configuration Service Setup](#)
- [No Configuration Service Setup – Application Ready for Use](#)
- [Navigation Configuration](#)
- [Customizing Application Branding](#)
- [Customizing Application Skins](#)
- [Customizing String Resources](#)
- [Resource Bundles](#)
- [Configuring Application for iOS Deployment](#)
- [Deploying Application to Distribution Package to Generate .apk File](#)
- [Android Scanning Support](#)
- [Details on Item Images](#)

Store Operations Cloud Service

Prerequisites

JDeveloper

Deploying Oracle Store Operations Cloud Service (SOCS) requires JDeveloper 12.2.1.3.

<https://www.oracle.com/tools/downloads/jdev-v12213-downloads.html>

Mobile Application Archive File

Mobile Application Archive file .maa can be downloaded from <https://edelivery.oracle.com> for the appropriate version of Store Operations Cloud Service (SOCS).

iOS Platform

Deploying Oracle SOCS for use on an iOS device requires that you have a computer running Mac OS X set up for iOS development. For more information on the setup, including provisioning profiles and certificates. Refer to Apple's documentation at <https://developer.apple.com/>.

Additionally, you must have installed JDeveloper, Xcode and Oracle MAF extension.

Also refer to *Oracle Mobile Application Framework* documentation for [Setting Up Development Tools for the iOS Platform](#).

Android Platform

Deploying Oracle SOCS for use on an Android device requires that you have a computer that has the Android SDK installed.

Additionally, you must have installed JDeveloper and Oracle MAF extension.

Also refer *Oracle Mobile Application Framework* documentation for [Setting Up Development Tools for the Android Platform](#).

Platform Specific Requirements

Build Time Support (SDK & build tool versions)

Mobile Platform	Mobile SDK Version	Native Development Tools Version
Android	Android 11.0 (API 30)	Minimum versions: Android SDK Tools 23, Platform-tools 27, Build-tools 27. If your MAF application uses a third-party Cordova plugin and the plugin uses native libraries for Android, ensure that the plugin also supports the 64-bit architecture for Android.
iOS	iOS SDK 15.x	Xcode 13.x Provisioning files and certificates For more Apple documentation see: https://developer.apple.com/ .

Installer Artifacts

File	Contents
maf-2.6600.20220729-0726.zip	This is the MAF extension that must be installed in JDeveloper (see Installing the MAF Extension below). MAF extension for JDeveloper can be found here: https://www.oracle.com/webfolder/technetwork/jdeveloper/downloads/1213center.xml#oracle.maf .
OracleSocsArchive.maa	The main SOCS application archive. (Required to install or customize Oracle SOCS)

Installing the MAF Extension

To install the MAF extension on already installed JDeveloper, see [Installing Mobile Application Framework with JDeveloper](#)

When you reach "Installing the MAF Extension in JDeveloper", click the **Install from Local File** radio button and then locate and select the maf-2.6600.20220729-0726.zip file (from [Installer Artifacts](#) above). Then skip to step 5.

Configuring JDeveloper for iOS Deployment

To configure your environment for the iOS platform:

1. Ensure Xcode and iOS SDK are installed.
2. In JDeveloper, click **Tools**, and then click **Preferences**.
3. In the Preferences dialog, click **Mobile Application Framework** and then click **iOS Platform**.
4. Enter the signing information and export options.

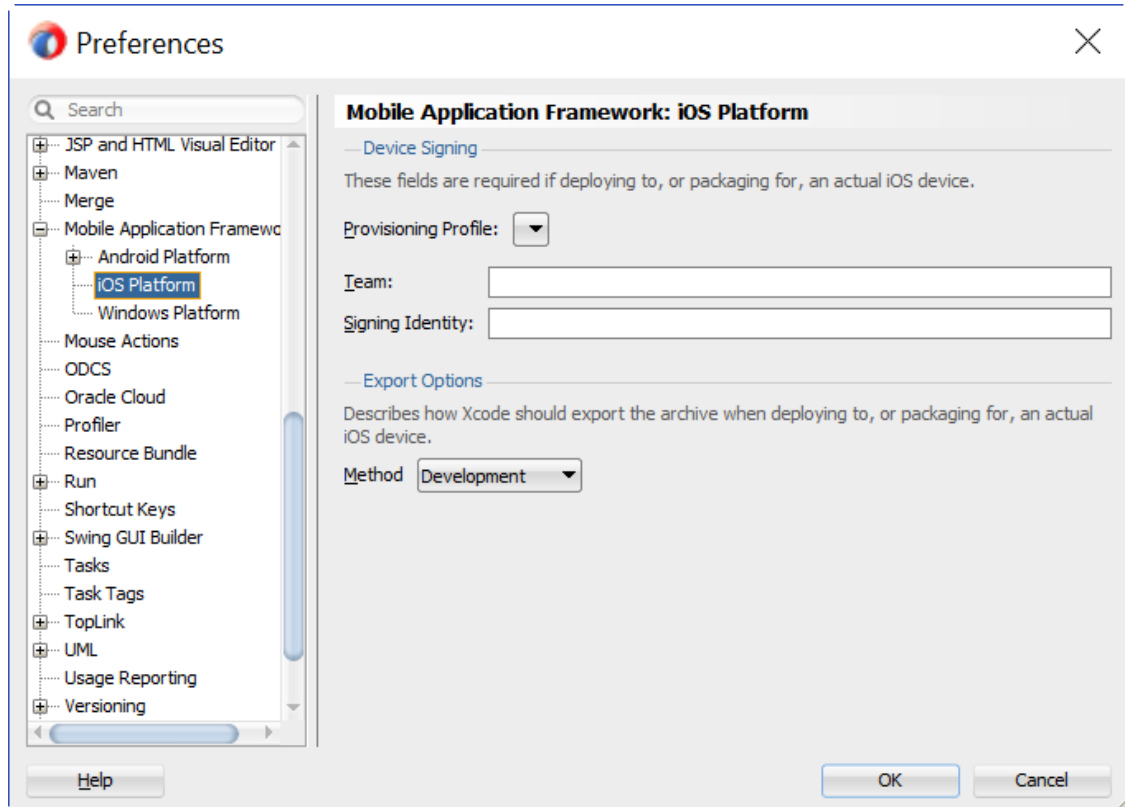
For more information, see [How to Set Device Signing and Export Options](#) in *Developing Mobile Applications with Oracle Mobile Application Framework*.

 Note:

Customers must use their own provisioning profile and signing identity to distribute the application. And Export Option method should be selected as for Release.

5. In the first box, use the browse button to select the **Automatically Add to iTunes** directory.
6. Select the **Provisioning Profile** from the drop-down list.
7. Enter the name of the signing identity corresponding to your development certificate.

Figure 2-1 Device Signing and Export Options of the iOS Platform Preferences



Configuring JDeveloper for Android Deployment

To configure your environment for the Android platform:

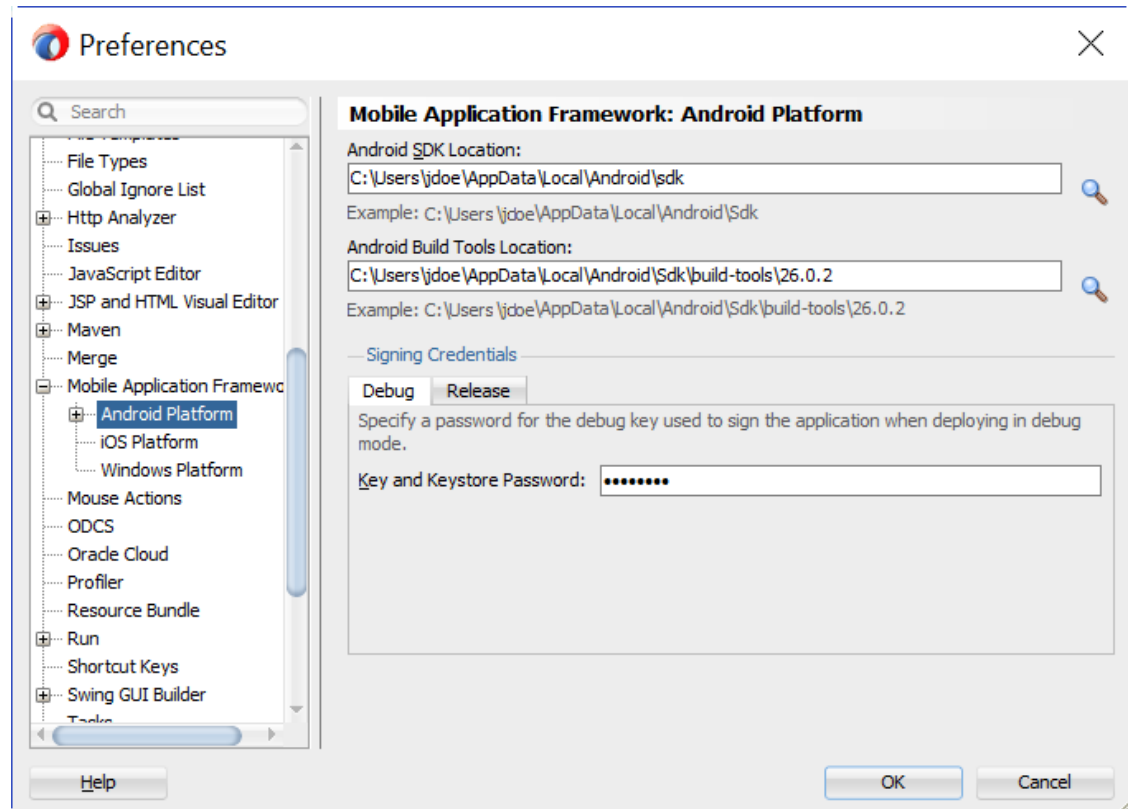
1. Ensure the Android SDK is installed.
2. In JDeveloper, click **Tools**, and then click **Preferences**.
3. In the Preferences dialog, click **Mobile Application Framework**, and then click **Android Platform**.
4. Specify the location of the Android SDK on your computer.
5. Specify the location of the Android build tools on your computer.
6. Provide information on the signing credentials.

For more information about creating a key and keystore, see [How to Sign a MAF Application that You Deploy to the Android Platform](#) in *Developing Mobile Applications with Oracle Mobile Application Framework*.

Note:

For performance reasons it is recommended to use Release in **Signing Credentials** section unless the intent is to debug the application.

Figure 2-2 Configuring Platform Preferences for Android



Creating a Workspace from the Archive

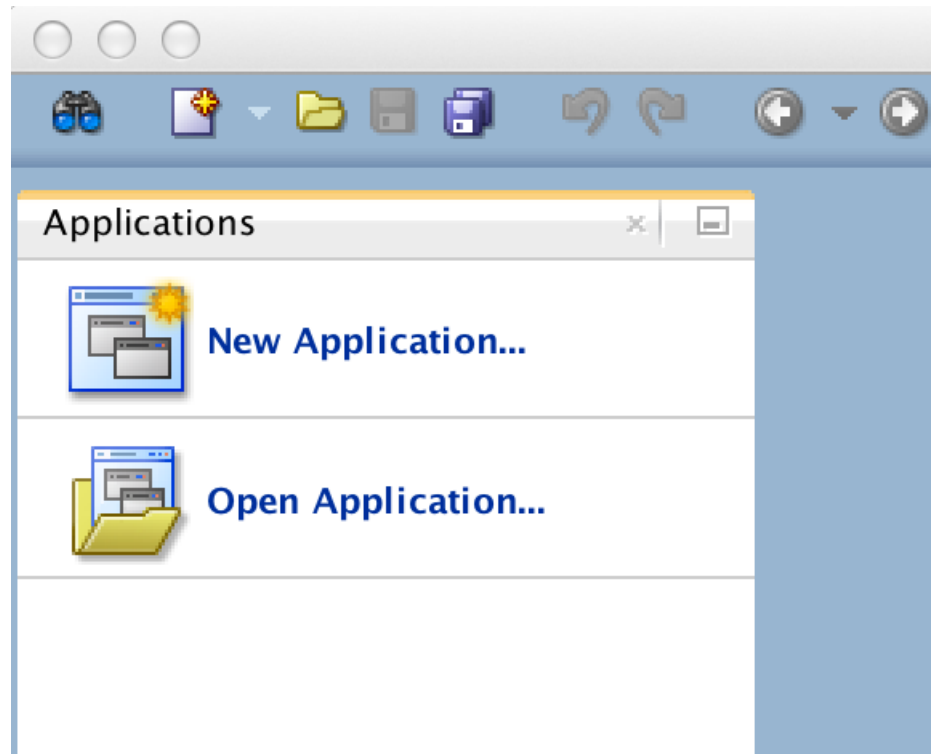
A mobile application archive (MAA) must be downloaded, then configured and built for the appropriate target operating system.

Any endpoint configuration or customization must be done prior to deployment.

When you launch JDeveloper for the first time, you will encounter an empty workspace. The first thing that must be done is to create a new application from the provided .maa file. Use the steps that follow:

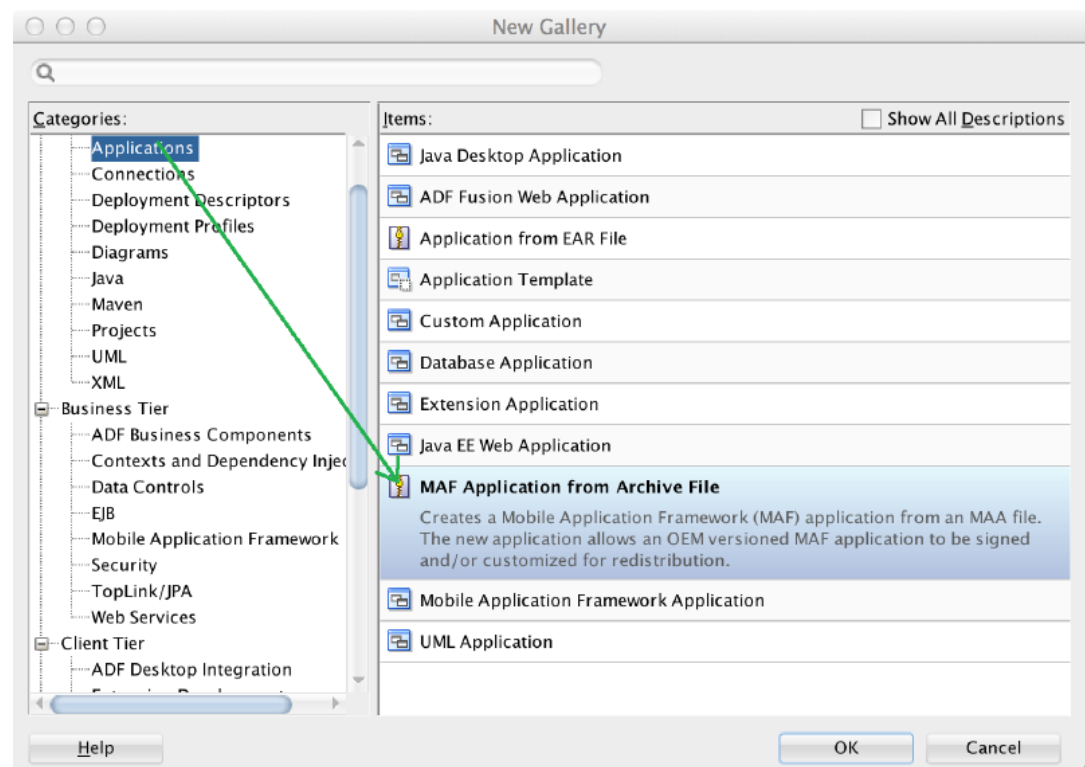
1. Open the **New Application** dialog (or open the general **New** dialog and select **Applications**).

Figure 2-3 New Application



2. Select **MAF Application** from Archive File and click **OK**.

Figure 2-4 MAF Application from Archive File



3. In the wizard that opens, select the .maa file you want to create a SOCS application from (the main Oracle SOCS app is packaged as OracleSocsArchive.maa).
4. Enter a name for your application.
5. You can also change where the application will be saved (by default, <workspace>/<application name>).
6. On the **Location** page, enter the location information, for example:
 - **MAA file:** /Users/Myuser/Desktop/OracleSocsArchive.maa
 - **Application file:** OracleSocsCustom
 - **Directory:** /Users/Myuser/jDeveloper/mywork/OracleSocsCustom
7. Click **Finish** when you have entered the necessary information.

For more information see [Creating a New Application from an Application Archive](#).

Configuring Application Settings

This section describes the configuration that is required (and some that is optional) prior to building and deploying the Oracle SOCS application.

Definitions of SOCS Configuration Service End Points

Configuring the application includes Configuring Service End Points mainly for allowing the application to authenticate users, connect to web services, and in certain cases, access remote images.

Feature	Connection (Type)
Configuration	ConfigServiceLogin (Login)
	ConfigService (URL)
Oracle SOCS	SimMobileLogin (Login)
	SimMobileService (URL)

The Oracle SOCS app uses Oracle IDCS (Identity Cloud Service) or Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) as its identify provider (IDP) for authentication. Several URLs and IDs will need to be configured prior to deployment for successful connection to the servers. Mainly:

- **oauthClientId** - is the client id of SOCS application provisioned by Oracle.
- **security.idcs.url** - is the end point url of **IDCS** provided by Oracle.
- **deploy.mobile.server.base.url** - is the end point url of base SOCS for ConfigService.
- **deploy.mobile.server.service.url** - is the end point url of SOCS server service for SimMobileService.

Oracle SOCS is delivered with placeholder connections for each of the connections listed above. It is possible to build and install the application without updating the included connections and then use the Configuration feature to update the connections after the application is installed on a device. For more information, see [Configuration Service Setup](#).

Another option is to update the placeholder connections with valid URLs prior to building the application so that the application is ready to run immediately after installation on a device. In either case, it is always possible to update the connections used on the device later through

the Configuration feature. For more information, see [No Configuration Service Setup – Application Ready for Use](#).

Figure 2-5 Edit MAF Login Connection.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<References xmlns="http://xmlns.oracle.com/adf/jndi">
  <Reference name="ConfigServiceLogin" className="oracle.adf.model.connection.adfmf.LoginConnection"
    adfCredentialStoreKey="ConfigServiceLogin" partial="false" manageInOracleEnterpriseManager="true"
    deployable="true" xmlns="">
    <Factory className="oracle.adf.model.connection.adfmf.LoginConnectionFactory"/>
    <RefAddresses>
      <XmlRefAddr addrType="adfmfLogin">
        <Contents>
          <oauthAuthorizationGrantType value="code"/>
          <oauthClientId value="@deploy.mobile.client.oauth.client.id"/>
          <oauthClientSecret/>
          <oauthRedirectEndpoint url="idcsmobileapp://nodata"/>
          <embeddedBrowser value="true"/>
          <oauthScopes>
            <oauthScope url="openid"/>
          </oauthScopes>
          <openIDConnectDiscoveryURL url="@security.idcs.url/./well-known/idcs-configuration"/>
          <authenticationMode value="remote"/>
          <accessControl/>
          <userObjectFilter/>
        </Contents>
      </XmlRefAddr>
    </RefAddresses>
  </Reference>
  <Reference name="ConfigService" className="oracle.adf.model.connection.url.HTTPURLConnection"
    adfCredentialStoreKey="ConfigServiceLogin" xmlns="">
    <Factory className="oracle.adf.model.connection.url.URLConnectionFactory"/>
    <RefAddresses>
      <XmlRefAddr addrType="ConfigService">
        <Contents>
          <urlconnection name="ConfigService" url="@deploy.mobile.server.base.url"/>
        </Contents>
      </XmlRefAddr>
    </RefAddresses>
  </Reference>
  <Reference name="SimMobileLogin" className="oracle.adf.model.connection.adfmf.LoginConnection"
    adfCredentialStoreKey="SimMobileLogin" partial="false" manageInOracleEnterpriseManager="true"
    deployable="true" xmlns="">
    <Factory className="oracle.adf.model.connection.adfmf.LoginConnectionFactory"/>
    <RefAddresses>
      <XmlRefAddr addrType="adfmfLogin">
        <Contents>
          <oauthAuthorizationGrantType value="code"/>
          <oauthClientId value="@deploy.mobile.client.oauth.client.id"/>
          <oauthClientSecret/>
          <oauthRedirectEndpoint url="idcsmobileapp://nodata"/>
          <embeddedBrowser value="true"/>
          <oauthScopes>
            <oauthScope url="openid"/>
          </oauthScopes>
          <openIDConnectDiscoveryURL url="@security.idcs.url/./well-known/idcs-configuration"/>
          <authenticationMode value="remote"/>
          <accessControl/>
          <userObjectFilter/>
        </Contents>
      </XmlRefAddr>
    </RefAddresses>
  </Reference>
</References>
```

For more information on accessing and configuring login connections, see [Configuring End Points Used in MAF Applications](#).



Note:

These updates must be made prior to building the application and cannot be updated through the Configuration feature.

Configuration Service Setup

If you are using the Configuration feature to update the `connections.xml` file on a mobile device after the application has been installed, it is necessary to host the `connections.xml` file at a secured location.

The hosted `connections.xml` file should contain valid URLs for all connections being used by the application (including the `ConfigService` and `ConfigServiceLogin` connections).

The `connections.xml` file must be named `connections.xml` and be located at a path that ends with the application identifier. For example, if the application identifier is `com.company.SOCS`, the `connections.xml` could be located at the following URL:

```
http://server:port/SomeLocation/com.company.socs/connections.xml
```

You need two URLs to complete the configuration process:

- **Configuration URL:** This is the base URL up to, but not including the application identifier. For example:

```
http://server:port/SomeLocation
```

 Note:

That this is also the value that can be pre-populated in the `ConfigService` connection.

- **Configuration Login URL:** This is a complete URL to a secured resource on the same domain as the `connections.xml` file. For example:

```
http://server:port/SomeLocation/SomeSecuredResource
```

 Note:

That this is also the value that can be pre-populated in the `ConfigServiceLogin` connection.

If the application is deployed with placeholder (non-valid URLs) in the `connections.xml` file and the user is required to use the Configuration feature to get valid connections, it is recommended that you configure the application to prompt the user to set up configuration on the initial launch of the application (until configuration has been completed).

To configure the application to prompt the user:

1. Locate and open the `adf-config.xml` file in JDeveloper under Application Resources / Descriptors / ADF META-INF.
2. In the source view, locate the `adf-property` tag with the name value `RETAIL_INITIAL_CONFIGURATION_REQUIRED`.
3. Change the value to `true`.

No Configuration Service Setup – Application Ready for Use

If the application is deployed with placeholder (non-valid URLs) in the connections.xml file and Oracle provides the connection.xml URL for SOCS, the user can build application so that the application is ready to run immediately after installation on a device.

To configure the application with pre-defined connections:

1. Locate and open the `adf-config.xml` file in JDeveloper under Application Resources / Descriptors / ADF META-INF.
2. In the source view, locate the `adf-property` tag with the name value `RETAIL_INITIAL_CONFIGURATION_REQUIRED`.
3. Change the value to false.
4. Modify the connection.xml in JDeveloper or located in `<SOCS workspace>/adf/META-INF/connection.xml` by replacing the below strings:
 - a. Replace all occurrences of `@deploy.mobile.client.oauth.client.id@` with SOCS Client ID from IDCS.
 - b. Replace all occurrences of `@security.idcs.url@` with the IDCS url.
 - c. Replace `@deploy.mobile.server.base.url@` with `https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/siocs-client-services`
 - d. Replace `@deploy.mobile.server.service.url@` with `https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/siocs-client-services/internal`

Or just copy the connection.xml contents provided by Oracle and replace that with the existing contents in the connection.xml in JDeveloper. Or in connection.xml located `<SOCS workspace>/adf/META-INF/connection.xml` in the Archive file workspace created by JDeveloper.



Note:

The `<Region Name>` and `<Customer Subnamespace>` part of the URL should be replaced with the one specific to your environment. This will be the same as your cloud service Application URL provided in the Welcome email.

Navigation Configuration



Note:

There is no need to update the navigation configuration unless changes in organization, labelling, and so on are desired.

The springboard is configured by a navigation.json file in the mobile application controller project. The file has the following structure:

```
{
```

```
menu : {  
  name : <Menu Title>,  
  taskFlowId : <ID of the task flow>  
  permission : <permission key that controls menu visibility>  
  options : [  
    {  
      name : <Feature group name>,  
      options : [  
        {  
          name : <Feature name>,  
          taskFlowId : <ID of the task flow>  
          permission : <permission key that controls menu visibility>  
          options : <Optional submenu>  
        },  
        ...  
      ]  
    },  
    ...  
  ]  
}  
footers : [  
  {  
    name : <Label for footer item>,  
    featureId : <ID of feature to link to>  
    taskFlowId : <ID of the task flow>  
  },  
  ...  
]
```

The bundles section allows the declaration of aliases to XLIFF bundles. Translated strings can then be used for name/label with [Alias.Key], similar to the usage in AMX pages (after using loadBundle). If an alias is not declared, the full bundle path must be used instead of the alias.

The menus section begins with the root node of the hierarchy. The name of this node is displayed on the switcher page of the springboard. The nodes under it are the feature groups to display on the switcher page. Finally, the next level of nodes are the features within the

group. If no name is provided, the name of the feature referenced is used. If the options property is specified, another level of feature nodes can be placed in it to form a submenu (the springboard does not support nesting submenus), and name is required in this case. Tapping on a feature item in the group or a submenu allows you to navigate to the provided feature identifier.

The footers section allows for setting up feature links that should be displayed regardless of current feature group. A feature link is included by specifying the featureId and optionally a label.

Customizing Application Branding

The application branding consists of the application icon as shown in various contexts and the splash screens when launching the application. The application branding can be customized by replacing the icons and images that are referenced by the application.

The following steps assume that a workspace has already been created from the Oracle SOCS MAA file and that the appropriate-sized icons or images have been added to the workspace created from the MAA file:

1. Open JDeveloper in the Studio Developer role.
2. Follow the steps outlined in *Developing Mobile Applications with Oracle Mobile Application Framework* for more information on How to Add a Custom Image.

[How to Add a Custom Image to an Android Application](#)

[Adding a Custom Image to an iOS Application](#)



Note:

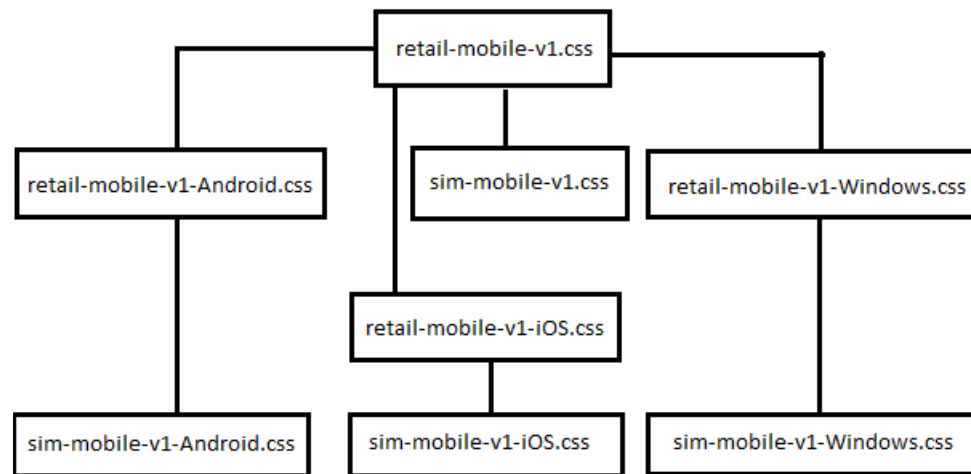
Since MDS is not aware of changes made to this image configuration, any customizations are overridden by upgrading to a newer version of Oracle SOCS.

Customizing Application Skins

The overall look and feel of the Oracle SOCS application is controlled by a skin. Since MAF supports MDS customizations of the maf-skins.xml and maf-config.xml files, it is possible to apply a custom skin to the application. If the Oracle SOCS application is upgraded to a newer version later, the skinning customizations are still applied on top of the upgraded version by MDS.

The following steps assume that a workspace has already been created from the Oracle SOCS MAA file:

1. Open JDeveloper in the Studio Developer role.
2. Open the Oracle SOCS workspace created from the delivered MAA file.
3. The currently used skinning files for Oracle SOCS are found in the SIMApplicationController project, inside of the public_html > platform > css subdirectory. The current hierarchy of files is represented below:



4. Create a new CSS skin file in the SIMApplicationController project. For more information on how to create CSS skin files and how MAF skinning works, see Chapter 9, [Skinning MAF Applications](#), in *Developing Mobile Applications with Oracle Mobile Application Framework*.
5. Switch JDeveloper to the Customization Developer role by selecting the Tools menu and then Switch Roles / Customization Developer.
6. When JDeveloper has restarted in the Customization Developer role, open the maf-skins.xml file.
7. In the Structure Pane, right click the adfmf-skins node and select Insert Inside adfmf-skins > skin.
8. Fill in the fields in the popup. Note that the style-sheet-name should reference the CSS skin file created in Step 3.
9. If the skin should be versioned, in the Structure Pane, right click the skin element just created, and select Insert Inside skin / version.
10. Fill out the popup with the skin version information. For more details on configuring skins in maf-skins.xml, see Chapter 9, [Skinning MAF Applications](#), in *Developing Mobile Applications with Oracle Mobile Application Framework*.
11. Save the changes.
12. Open the maf-config.xml file.
13. Update the skin-family element with the name of the custom skin family that should be applied to the entire application.
14. Update the skin-version element if the custom skin was defined with a different version. If the custom skin does not have a version, remove the skin-version element.
15. Save the changes.
16. Deploy the application to iOS Simulator to verify that the new skin is being picked.

After upgrading to a new Oracle SOCS version, the custom skin changes are still applied over the base application.

Customizing String Resources

In order to support localization, Oracle SOCS references application string resources in the XLIFF resource bundles.

Since Oracle MAF does not support MDS customizations of XLIFF resource bundles, the Customization Developer role does not allow you to make changes to the XLIFF resource bundles. Any changes made in the Studio Developer role to the XLIFF resource bundles delivered in the Oracle SOCS application are overridden when the application is upgraded at a later date, so this approach is not recommended.

In order to add custom string resources to the Oracle SOCS application in a future-proof way, you must create new resource bundles under the Studio Developer role, add strings to these new bundles, and then reference strings from these new resource bundles in customizations to AMX pages or other customizable artifacts under the Customization Developer role. For more information about this process, see *Enabling Customizations in Resource Bundles* in *Developing Mobile Applications with Oracle Mobile Application Framework*.

Resource Bundles

The following table lists the resource bundles that are packaged in the Oracle SOCS MAA file:

Base XLIFF Resource Bundle	Text Resource Descriptions
OracleSOCS/.adf/META-INF/SimMobileBundle.xlf	Text resource for the application name.
OracleSOCS/SimMobileViewController/src/oracle/retail/sim/mobile/SimMobileViewControllerBundle.xlf	Text resources for the Oracle SOCS feature.
OracleSOCS/SimApplicationController/src/oracle/retail/sim/mobile/client/application/SimMobileApplicationControllerBundle.xlf	Text resources for the application names as shown on the springboard.

Configuring Application for iOS Deployment

To configure Oracle SOCS for iOS deployment, use the following steps:

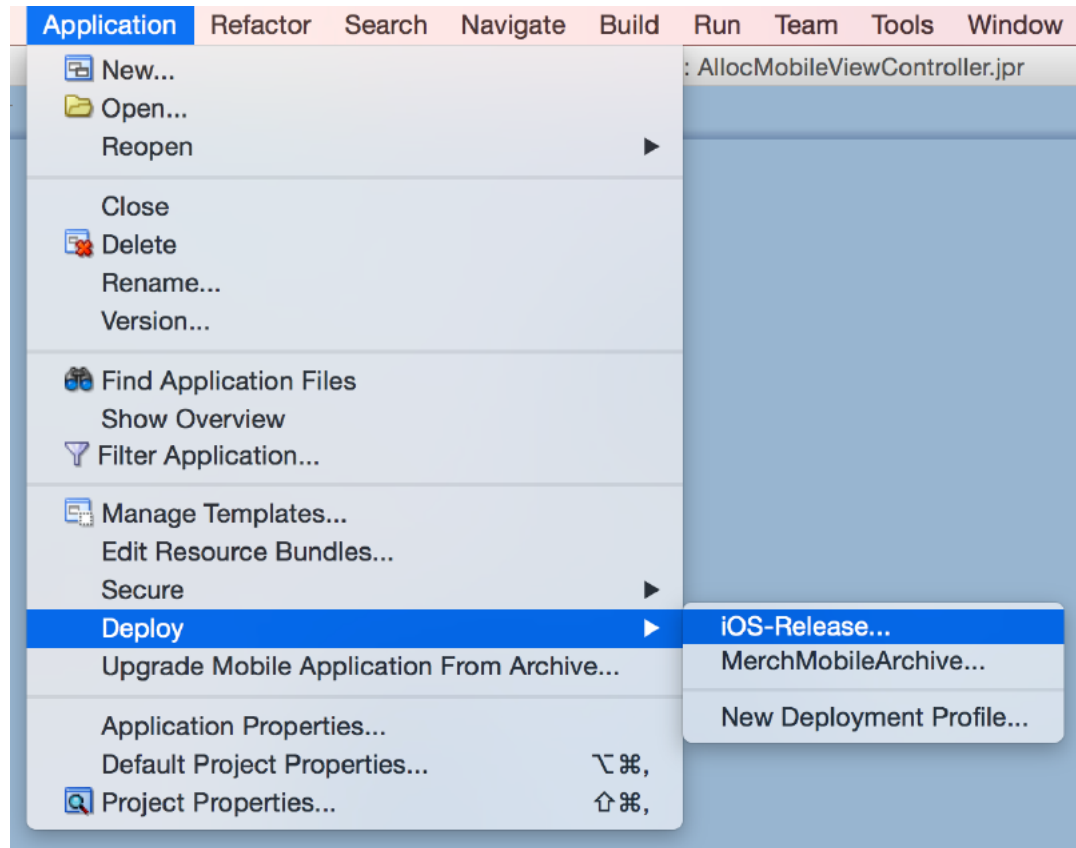
1. Open the workspace created above in [Creating a Workspace from the Archive](#).
 - a. Open the application properties (Application' Application Properties...).
 - b. Under **Deployment**, select the iOS1 deployment profile and click the pencil icon.
2. Select **iOS Options** and confirm you have the correct Application Bundle Id and Application Archive Name (for more information about application bundle ids, refer to *Oracle Fusion Middleware Developing Mobile Applications with Oracle Mobile Application Framework* documentation.)
3. Confirm that the application deploys in Release Mode.
4. Check the other settings as needed.
5. Click **OK**.

Deploying Application to Distribution Package to Generate .ipa File

To deploy Oracle SOCS to a .ipa file, use the following steps:

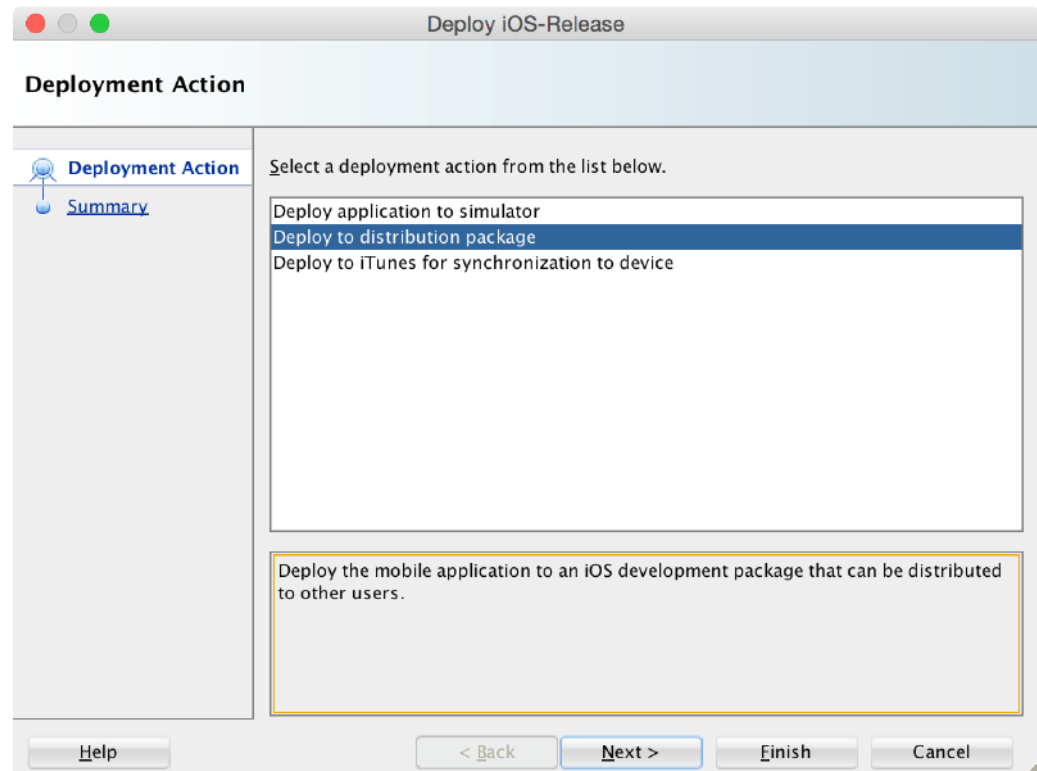
1. In the Application menu, select **Deploy/iOS-Release**.

Figure 2-6 Deploy iOS-Release



- a. Select **Deploy to distribution package**.
- b. Press **Finish** to begin deployment.

Figure 2-7 Deployment Action - Deploy iOS-Release



2. You can view the progress of the deployment by opening the Log window. (Window' Log).
3. After a successful deployment you can find the .ipa file within your workspace under the deploy folder.

For Enterprise app distribution refer to:

https://developer.apple.com/documentation/xcode/preparing_your_app_for_distribution

For other distribution options, see *Oracle Fusion Middleware Developing Mobile Applications with Oracle Mobile Application Framework* documentation.



Note:

If Oracle SOCS is already deployed to a device, it is recommended to delete it from the device before redeploying it again.

Configuring Application for Android Deployment

To configure Oracle SOCS for Android deployment, use the following steps:

1. Open the workspace created above in [Creating a Workspace from the Archive](#).
 - a. Open the application properties (Application ' Application Properties...).
 - b. Under Deployment, select the Android1 deployment profile and click the pencil icon.
2. Select **Android Options** and confirm you have the correct Android settings.

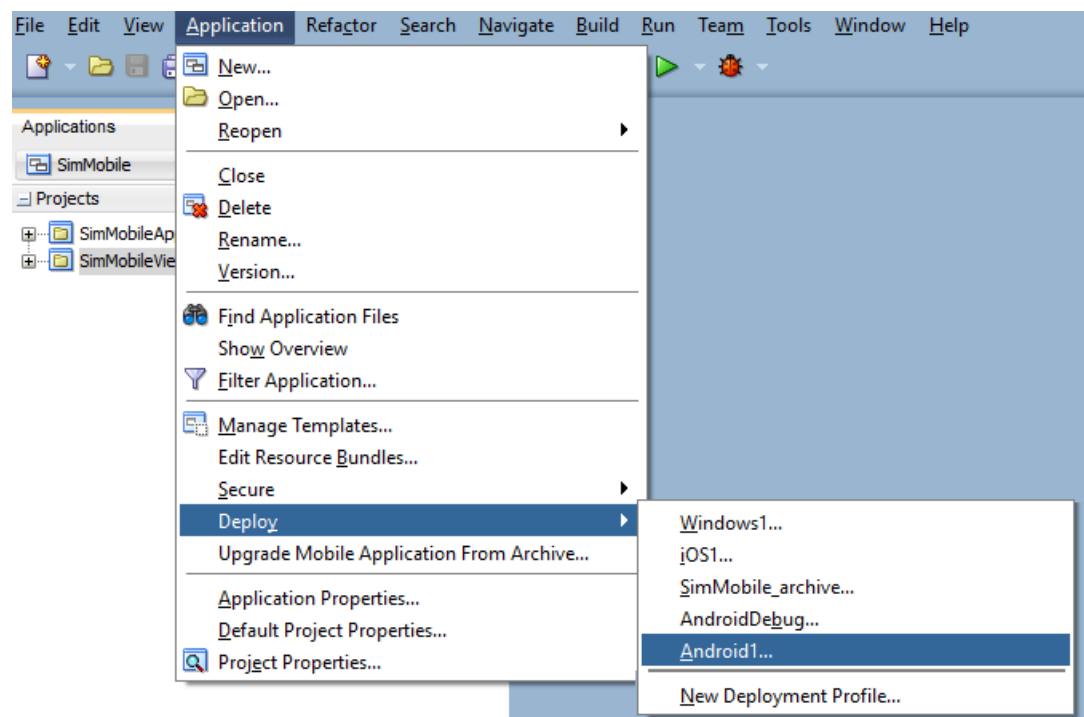
3. Confirm that the application deploys in Release Mode.
4. If you wish to change Android application settings, do so in the subsections Application Details and Application Images.
5. Check the other settings as needed.
6. Click **OK**.

Deploying Application to Distribution Package to Generate .apk File

To deploy Oracle SOCS to a .apk file, use the following steps:

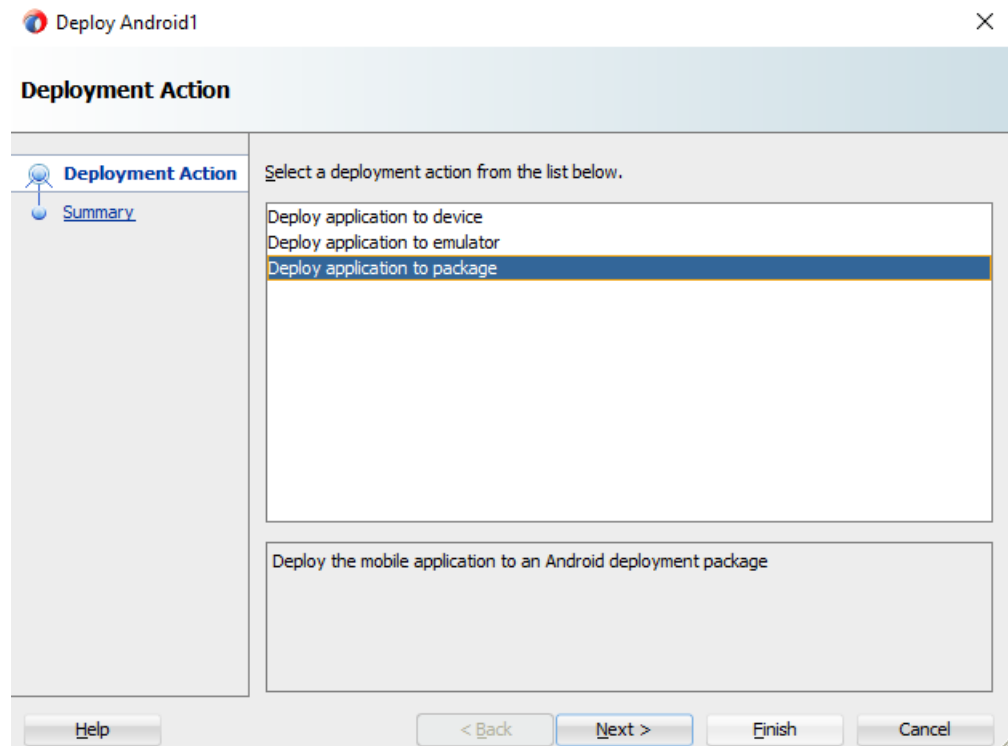
1. In the Application menu, select **Deploy/Android1**.

Figure 2-8 Deploy - Android1



- a. Select **Deploy Application to package**.
- b. Press **Finish** to begin deployment.

Figure 2-9 Deployment Action - Deploy Android



2. You can view the progress of the deployment by opening the Log window. (Window' Log)
3. After a successful deployment you can find the .apk file within your workspace under the deploy folder.

 Note:

If Oracle SOCS is already deployed to a device, it is recommended to delete it from the device before redeploying it again.

Android Scanning Support

Oracle SOCS is already deployed to a device; it is recommended to delete it from the device before redeploying it again.

Configuring DataWedge Profile for Scanning

 Note:

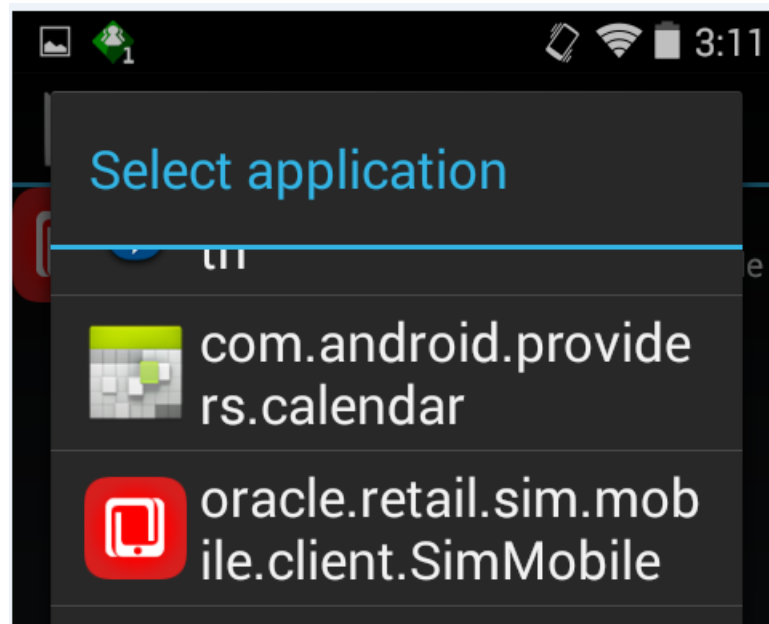
This is applicable to Honeywell /Zebra /Symbol / Motorola Android devices only.

Open the DataWedge application on the device.

1. Click on the **Menu Bar** at the top right corner. Select **New Profile**.

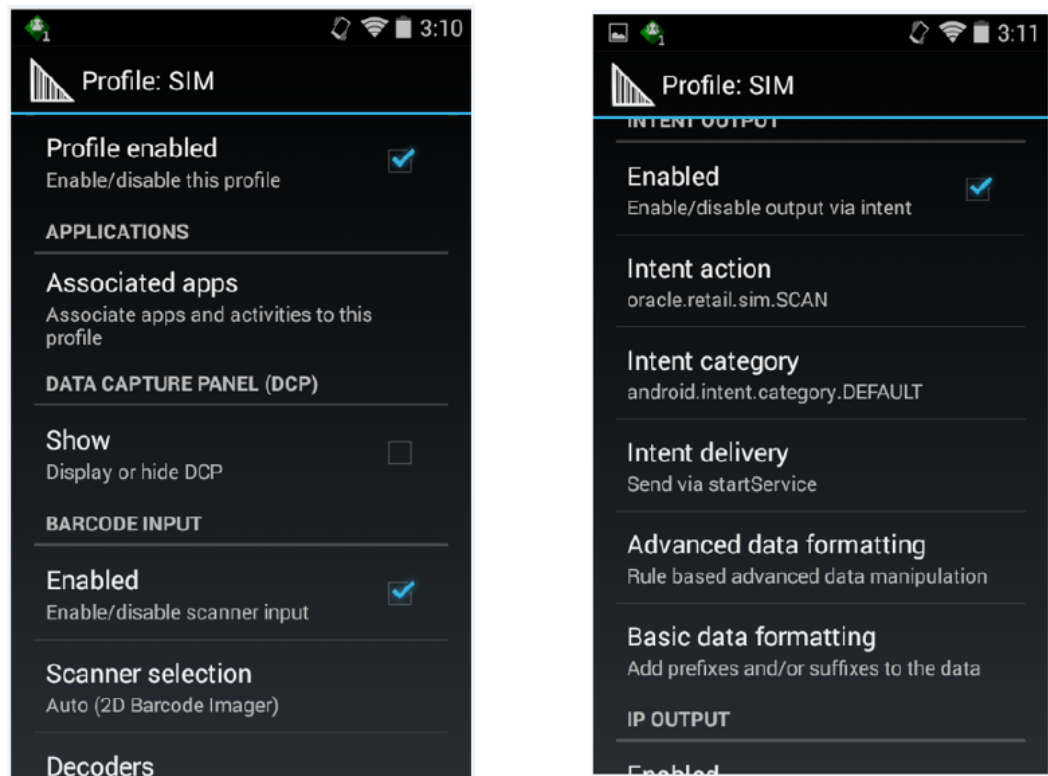
- a. Enter the profile name.
 - b. Select the newly created profile.
2. Make sure only these are checked:
 - a. Profile enabled - enable/disable this profile
 - b. Barcode Input - enable/disable scanner input
 - c. Intent Output - enable/disable output via intent
3. Under the APPLICATIONS group select **Associated Apps**:

Figure 2-10 Associated Apps



- a. Select the oracle.retail.sim.mobile.client.SOCSSMobile identifier.
4. Under the INTENT OUTPUT set:
 - a. Intent action to oracle.retail.sim.SCAN
 - b. Intent category to android.intent.category.DEFAULT
 - c. Intent delivery to Send via Broadcast Intent
5. Save the settings and you would be able use the hardware buttons to provide scan inputs to the Oracle SOCS application.

Figure 2-11 Profile Settings



Details on Item Images

If configured, the SOCS mobile application will display images for items that have been entered into the system. Different sizes of images are supported in the application for optimal performance. In addition, the application supports a configurable amount of background threads to further improve image loading speeds.

Image Sizing

There are two different image sizes used in the SOCS mobile application. These sizes exist to improve performance where different sizes of images are displayed by not using excess bandwidth (for example high resolution images being loaded and scaled down to a small space).

Thumbnail Images

Thumbnail images are the main images utilized in the application. These images are intended for areas in the application where smaller images are displayed (for example lists of items). The suggested resolution of these images is about 128x128px to 256x256px depending on the resolution and DPI of the mobile device's screen running the application. Any higher resolution images are scaled down, and the increased bandwidth to load the images is merely wasted and can cause decreases in performance. Thumbnail images are loaded asynchronously after the screen is populated, and there is a configurable amount of background threads to increase the rate multiple images are loaded into the application.

High Resolution Images

High resolution images, denoted by the size code 'H' in the system, are loaded on demand by a user. These images are meant to be large up to full screen images of an item. Thus, the suggested resolution of these images is up to the screen resolution of the device. These images can be 600x600 pixels to 800x800 pixels or even greater.

Configurable Threads

The SOCS mobile application also supports a configurable number of background threads to load multiple images at once. The configured number of threads should be an appropriate number for the mobile hardware running the SOCS application. An excess number of threads can severely impact performance of the device.

3

Installing and Setting up the Application - JET Mobile

Prerequisites

You must have downloaded the Retail Store Operations Cloud Service package from OSDC (edelivery.oracle.com) and extracted the apk file from the /SOCS-JET folder.

You must have a separate IDCS application provisioned:

Format: RGBU_SIOCS_< EnvType><EnvIndex >_SOCS_JETMOBILE

Example: RGBU_SIOCS_DEV1_SOCS_JETMOBILE

Platform-Specific Requirements

End User Device Android OS Version

Android	10.0+
---------	-------

Pre-Installation Notes

For the Android builds (APKs) of the JET Mobile Application there are versioned and standard builds. Only one standard build can be installed on a device at any given time, as all standard builds will have the same Android bundle identifier. Versioned JET mobile builds have their version numbers included in their Android bundle identifiers so multiple versions of the “same” app can be installed on a device at any given time. Example bundle identifiers are:

Standard: oracle.retail.sim.mobile.client

Versioned: oracle.retail.sim.mobile.client.v250101

In addition to different bundle identifiers, standard and versioned APK builds will have different names appear on the operating system – where the versioned build will have the version as part of its app name. Examples:

Standard: Oracle SOCS

Versioned: SOCS251010



Note:

The above section pertains to Android JET mobile builds only. For Apple iOS builds, customers have the ability to specify their own application name and unique bundle identifier (see [Building JET Mobile as an iOS Application](#)).

Configuring Application Settings

This section describes the configuration that is required prior to using the JET Mobile application.

Definitions of JET Mobile

Configuring the application includes Configuring Service End Points mainly for allowing the application to authenticate users, and connect to web services.

The client application configuration can be input manually using the UI that will appear after the EULA is displayed if the configuration is empty or incomplete.

Alternatively, the application configuration can be setup via an Android compatible MDM.

In both cases there are 3 configuration settings that are required.

When entering these settings using an MDM it is important to use the correct key and type for each value.

In Android, these MDM settings are referred to as Managed Configurations.

Base Service URI:

For example the full SIOCS service URI would be:

`https://rex.retail.<Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/siocs-client-services/internal`



Note:

The <Region Name> and <Customer Subnamespace> part of the URL should be replaced with the one specific to your environment. This will be the same as your cloud service Application URL provided in the Welcome email.

The base URI does not need to include the suffix “siocs-client-services/internal” but it will be removed if it is included.

MDM managed configuration properties:

Key: siocsBaseUri

Type: String

IDCS Base URI:

This is the base URI for the IDCS tenant.

MDM managed configuration properties:

Key: idcsBaseUri

Type: String

OAuth Client ID:

This is the IDCS public client app that was provisioned in IDCS for this mobile client application.

Format: RGBU_SIOCS_< EnvType><EnvIndex >_SOCS_JETMOBILE_APPID

Example: RGBU_SIOCS_DEV1_SOCS_JETMOBILE_APPID

MDM managed configuration properties:

Key: oauthClientId

Type: String

Configuring DataWedge Profile for Scanning



Note:

This is applicable to Honeywell / Zebra / Symbol / Motorola Android devices only.

1. Click on the **Menu Bar** at the top right corner. Select **New Profile**.
 - a. Enter the profile name.
 - b. Select the newly created profile.
2. Make sure only these are checked:
 - a. Profile enabled - enable/disable this profile
 - b. Barcode Input - enable/disable scanner input
 - c. Intent Output - enable/disable output via intent
3. Under the APPLICATIONS group select **Associated Apps**:
 - a. Select the oracle.retail.sim.mobile.client identifier.
4. Under the INTENT OUTPUT set:
 - a. Intent action to oracle.retail.sim.mobile.ZEBRA_SCAN (For ZEBRA Device)
Intent action to oracle.retail.sim.mobile. HONEYWELL_SCAN (For HONEYWELL Device)
 - b. Intent category to android.intent.category.DEFAULT
 - c. Intent delivery to Send via Broadcast Intent
5. Save the settings and you would be able use the hardware buttons to provide scan inputs to the JET mobile application.

Configuring Oracle MICROS devices for Scanning

1. Enter the scanner Output Method Settings:
 - a. Toggle on "Broadcast".
 - b. For "Broadcast of scan result" set the value to "oracle.retail.sim.mobile.MICROS_SCAN".
 - c. Set "Key Value" to "data".

- d. Set "Original Byte" to "source_byte".
2. Save the settings and restart the JET Mobile application. Any scan from the Oracle MICROS device should now automatically be sent to the app.

Building JET Mobile as an iOS Application

The SIOCS JET Mobile application is distributed for iOS as a .framework archive which needs to be imported and built into an iOS application that can then be distributed internally to customers' organizations.

Prerequisites

In order to build the JET mobile application, there are several prerequisites that need to be obtained from the Apple Developer organization.

- An Apple Developer account
- An Apple MacBook is necessary to build a native iOS app.
- Latest version of the XCode development application (v 15.4+)
- An application Provisioning Profile of type 'Enterprise' obtained through the Apple Developer account is necessary to distribute an app internally.
- An application Signing Certificate, also obtained through the Apple Developer program
- After XCode has been installed on the MacBook, the provisioning profile and certificate also need to be installed.

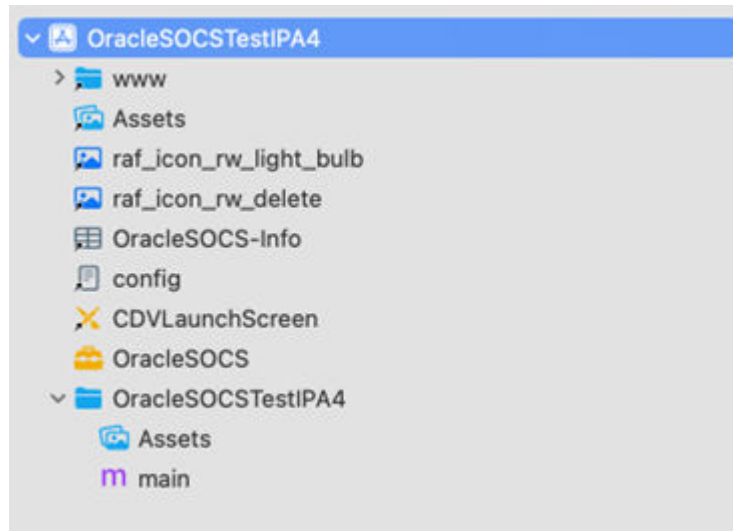
Building the Application

In XCode, a project must be created and configured to use the distributed JET Mobile .framework before being deployed. Follow the steps below:

1. Open XCode → New Project.
2. Select **iOS** option on top of dialog → **App** type project → Next.
3. Enter Product Name (of your choice) → Select your Developer Team → Enter custom bundle identifier of your choice (it needs to be unique).
4. Set Interface to **Storyboard** → Language to **Objective-C** → Storage to **None** → Hit Next.
5. Select directory for the project → Hit **Create**.
6. Copy the *OracleSOCS.framework* distributed by Oracle to the project directory.
7. Select click the project on the left panel → Under the General tab in main area → Select the build target → Frameworks, Libraries section → Hit + and select the *OracleSOCS.framework*.
8. Delete all generated files **except** for main.m and Assets folder.
9. Right click the project on the left panel → Add Files → From the *OracleSOCS.framework*: **CDVLaunchScreen.storyboardc**, **config.xml**, **OracleSOCS-Info.plist**, **raf_icon_rw_delete.png**, **raf_icon_rw_light_bulb.png**, and the "www" folder.
10. Select the project on the left panel, select the build target. Under the "Build Settings" category:
 - a. Under the "Packaging" subcategory enter "**OracleSOCS.framework/OracleSOCS-Info.plist**" for the "Info.plist File" entry.

- b. Under the "Info.plist Values" set the "Launch Screen Interface File Base Name" setting to **"CDVLaunchScreen"**.
 - c. Clear the entry of "UIKit Main Storyboard File Base Name".
11. Select click the "Assets" item under the project at the left panel and select "+" at the bottom → Import → Select OracleSOCS.framework/AppIcon.appiconset folder.
 - a. Note the name of the imported icon set. It's likely something like "AppIcon 1".
12. Select the project in the left pane, select its build target. Under General → App Icons and Launch Screen → Set "App Icon" to that value above (i.e. AppIcon 1).
13. This concludes the mandatory steps. Other settings can be customized like App Bundle ID, Application Name, etc.
14. Archive the project, and distribute the archive based upon your Apple Developer provisioning settings.
15. Your project should resemble the structure below:

Figure 3-1 Building the Application



Contents of main.m

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

#import <OracleSOCS/OracleSOCSAppDelegate.h>

int main(int argc, char * argv[]) { @autoreleasepool { return UIApplicationMain(argc, argv, nil,
@"AppDelegate"); }
```

Running JET Mobile as a Browser-Based Application

Jet Mobile can also be launched as a web application on Android, Windows and iOS platforms. Please note that web application does not support scanning.

For optimal user experience and efficiency, mobile devices with built-in scanners such as Zebra or Honeywell would still be preferred.

Browser based application can be launched as follows:

`https://rex.retail<.Region Name>.ocs.oraclecloud.com/<Customer Subnamespace>/siocs-mobile-web-client`

Details on Item Images

If configured, the JET mobile application will display images for items that have been entered into the system. Different sizes of images are supported in the application for optimal performance.

Image Sizing

There are two different image sizes used in the SOCS mobile application. These sizes exist to improve performance where different sizes of images are displayed by not using excess bandwidth (for example high resolution images being loaded and scaled down to a small space).

Thumbnail Images

Thumbnail images are the main images utilized in the application. These images are intended for areas in the application where smaller images are displayed (for example lists of items). The suggested resolution of these images is about 128x128px to 256x256px depending on the resolution and DPI of the mobile device's screen running the application. Any higher resolution images are scaled down, and the increased bandwidth to load the images is merely wasted and can cause decreases in performance. Thumbnail images are loaded asynchronously after the screen is populated, and there is a configurable amount of background threads to increase the rate multiple images are loaded into the application.

High Resolution Images

High resolution images, denoted by the size code 'H' in the system, are loaded on demand by a user. These images are meant to be large up to full screen images of an item. Thus, the suggested resolution of these images is up to the screen resolution of the device. These images can be 600x600 pixels to 800x800 pixels or even greater.

Network Considerations

Images are sent as a URL from the server to the JET mobile application. The JET mobile application webview then will load and display the image that corresponds to the URL the server provided. Therefore the end user devices need network access to any image URL endpoints for the images to be loaded and rendered.