Oracle® E-Business Suite

Mobile Apps Developer's Guide

Release 12.1 and 12.2

Part No. E69284-17

August 2025

ORACLE

Oracle E-Business Suite Mobile Apps Developer's Guide, Release 12.1 and 12.2

Part No. E69284-17

Primary Author: Vijay Shanmugam, Melody Yang

Contributing Author: Hadi Alatasi, Sugathan Aravindan, Prasanna Athota, Srinivasa Rao Atla, Rekha Ayothi, Eric Bing, Krishna Botta, Hubert Ferst, Rajesh Ghosh, Erik Graversen, Saritha Merugu, Lohit Moripalli, Ravindra Nadakuditi, Sanyukta Palod, Arun Purushothaman, Esteban Rodriguez, Dilbagh Singh, Ryoji Suzuki, Karthik Vase, Anjana Viswanath, Bill Wyza

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 customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Contents

4 Using Mobile Application Archives for Enterprise Distribution

5 Using the Login Component to Develop Mobile Apps

6 Implementing Push Notifications

Send Us Your Comments

Oracle E-Business Suite Mobile Apps Developer's Guide, Release 12.1 and 12.2

Part No. E69284-17

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 new Oracle E-Business Suite Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_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 www.oracle.com.

Preface

Intended Audience

Welcome to Release 12.1 and 12.2 of the *Oracle E-Business Suite Mobile Apps Developer's Guide.*

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.
- Computer desktop application usage and terminology.
- Oracle E-Business Suite applications.

This documentation assumes familiarity with Oracle E-Business Suite. It is written for the technical consultants, implementers and system integration consultants who oversee the functional requirements of these applications and deploy the functionality to their users.

If you have never used Oracle E-Business Suite, we suggest you attend one or more of the Oracle E-Business Suite training classes available through Oracle University.

See Related Information Sources on page xiv for more Oracle E-Business Suite product information.

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 customer access to and use of Oracle support services will be pursuant to the

terms and conditions specified in their Oracle order for the applicable services.

Structure

Related Information Sources

This book is included in the Oracle E-Business Suite Documentation Library. If this guide refers you to other Oracle E-Business Suite documentation, use only the latest Release 12.2 versions of those guides.

Online Documentation

All Oracle E-Business Suite documentation is available online (HTML or PDF).

- **Online Help** - Online help patches (HTML) are available on My Oracle Support.

- **Oracle E-Business Suite Documentation Library** - This library, which is included in the Oracle E-Business Suite software distribution, provides PDF documentation as of the time of each release.

- **Oracle E-Business Suite Documentation Web Library** - This library, available on the Oracle Help Center, provides the latest updates to Oracle E-Business Suite documentation. See https://docs.oracle.com/cd/E26401_01/index.htm for the latest Release 12.2 documentation or https://docs.oracle.com/cd/E18727_01/index.htm for the latest Release 12.1 documentation. Most documents are available in PDF and HTML formats.

- **Release Notes** - For information about changes in this release, including new features, known issues, and other details, see the release notes for the relevant product, available on My Oracle Support.

- **Oracle Electronic Technical Reference Manual -** The Oracle Electronic Technical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for each Oracle E-Business Suite

product. This information helps you convert data from your existing applications and integrate Oracle E-Business Suite data with non-Oracle applications, and write custom reports for Oracle E-Business Suite products. The Oracle eTRM is available as an application in Oracle E-Business Suite.

Related Guides

You should have the following related books on hand. Depending on the requirements of your particular installation, you may also need additional manuals or guides.

Oracle Alert User's Guide

This guide explains how to define periodic and event alerts to monitor the status of your Oracle E-Business Suite data.

Oracle Diagnostics Framework User's Guide

This manual contains information on implementing and administering diagnostics tests for Oracle E-Business Suite using the Oracle Diagnostics Framework.

Oracle E-Business Suite Concepts

This book is intended for those planning to deploy Oracle E-Business Suite Release 12.2, or make significant changes to a configuration. After describing the Oracle E-Business Suite architecture and technology stack, it moves on to give an outline of the actions needed to achieve a particular goal, plus any installation and configuration choices.

Oracle E-Business Suite CRM System Administrator's Guide

This manual describes how to implement the CRM Technology Foundation (JTT) and use its System Administrator Console.

Oracle E-Business Suite Developer's Guide

This guide contains the coding standards followed by Oracle E-Business Suite Development. It describes the Oracle Application Object Library components needed to implement the Oracle E-Business Suite user interface described in the *Oracle E-Business Suite User Interface Standards for Forms-Based Products*. It provides information to help you build your custom Oracle Forms Developer forms so that they integrate with Oracle E-Business Suite. In addition, this guide has information for customizations in features such as concurrent programs, flexfields, messages, and logging.

Oracle E-Business Suite Electronic Technical Reference Manual User's Guide

This guide describes how to set up and navigate Oracle E-Business Suite Electronic Technical Reference Manual (eTRM) user interface in Oracle E-Business Suite. It also explains how to browse and search the Oracle eTRM repository to locate desired FND and database metadata and objects, and how to view object details, reports, and diagrams.

Oracle E-Business Suite Maintenance Guide

This book explains how to patch an Oracle E-Business Suite system, describing the adop patching utility and providing guidelines and tips for performing typical patching operations. It also describes maintenance strategies and tools designed to help keep a

system running smoothly.

Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2

This guide includes the latest mobile release with new underlying technologies, as well as the earlier mobile releases built with Oracle Mobile Application Framework (MAF). It explains how to set up an Oracle E-Business Suite instance to support connections from Oracle E-Business Suite mobile apps. It also describes common administrative tasks for configuring Oracle E-Business Suite mobile apps. Logging and troubleshooting information is also included in this book.

Oracle E-Business Suite Integrated SOA Gateway User's Guide

This guide describes the high level service enablement process, explaining how users can browse and view the integration interface definitions and services residing in Oracle Integration Repository.

Oracle E-Business Suite Integrated SOA Gateway Implementation Guide

This guide explains how integration administrators can manage and administer the web service activities for integration interfaces including native packaged integration interfaces, composite services (BPEL type), and custom integration interfaces. It also describes how to set up and implement Service Invocation Framework to invoke SOAP and REST services from Oracle E-Business Suite, and how to manage web service security, configure logs, and monitor both inbound service invocations using Service Monitor and outbound service invocations through Service Invocation Framework using Service Invocation Monitor.

Oracle E-Business Suite Integrated SOA Gateway Developer's Guide

This guide describes how integration developers can perform end-to-end service integration activities. These include orchestrating discrete web services into meaningful end-to-end business processes using business process execution language (BPEL), and deploying BPEL processes at runtime.

This guide also explains how to invoke web services using the Service Invocation Framework. This includes defining web service invocation metadata, invoking web services, and testing web service invocation.

Oracle E-Business Suite Security Guide

This guide contains information on a comprehensive range of security-related topics, including access control, user management, function security, data security, secure configuration, and auditing. It also describes how Oracle E-Business Suite can be integrated into a single sign-on environment.

Oracle E-Business Suite Setup Guide

This guide contains information on system configuration tasks that are carried out either after installation or whenever there is a significant change to the system. The activities described include defining concurrent programs and managers, enabling Oracle Applications Manager features, and setting up printers and online help.

Oracle E-Business Suite User's Guide

This guide explains how to navigate products, enter and query data, and run concurrent requests by means of the user interfaces (UI) of Oracle E-Business Suite. It includes basic information on setting preferences and customizing the UI. An introduction to Oracle Enterprise Command Centers is also included. Lastly, this guide describes accessibility features and keyboard shortcuts for Oracle E-Business Suite.

Oracle E-Business Suite User Interface Standards for Forms-Based Products

This guide contains the user interface (UI) standards followed by Oracle E-Business Suite Development. It describes the UI for Oracle E-Business Suite products based on Oracle Forms, and how to apply this UI to the design of such applications.

Oracle Workflow Administrator's Guide

This guide explains how to complete the setup steps necessary for any product that includes workflow-enabled processes. It also describes how to manage workflow processes and business events using Oracle Applications Manager, how to monitor the progress of runtime workflow processes, and how to administer notifications sent to workflow users.

Oracle Workflow Developer's Guide

This guide explains how to define new workflow business processes and customize existing Oracle E-Business Suite-embedded workflow processes. It also describes how to configure message metadata for Oracle Mobile Approvals for Oracle E-Business Suite and how to define and customize business events and event subscriptions.

Oracle Workflow User's Guide

This guide describes how users can view and respond to workflow notifications and monitor the progress of their workflow processes.

Do Not Use Database Tools to Modify Oracle E-Business Suite Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle E-Business Suite data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle E-Business Suite data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle E-Business Suite tables are interrelated, any change you make using an Oracle E-Business Suite form can update many tables at once. But when you modify Oracle E-Business Suite data using anything other than Oracle E-Business Suite, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle E-Business Suite.

When you use Oracle E-Business Suite to modify your data, Oracle E-Business Suite automatically checks that your changes are valid. Oracle E-Business Suite also keeps

track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

Part 1

Oracle E-Business Suite Mobile Apps Release 10.x

1

Developer Considerations for Oracle E-Business Suite Mobile Apps Release 10.x

Overview

In Release 10.x, the Oracle E-Business Suite mobile apps uptake new underlying technologies that are no longer dependent on Oracle Mobile Application Framework. These new technologies such as Oracle JavaScript Extension Toolkit (JET) and Cordova framework allow developers to efficiently build client-side applications for cross-platform development easily.

> **Note: Oracle JET** is a collection of Oracle and open source JavaScript libraries, and this development toolkit allows you to efficiently build the cross-platform and client-side hybrid mobile apps based on JavaScript, HTML5, and CSS3. **Cordova framework** is also an open-source mobile development framework allowing you to build apps using standard web technologies such as HTML5, CSS, and JavaScript for cross-platform development. With either of these technologies, you can build an app that runs on both Android and iOS devices without having to use any platform-specific tools.

Oracle E-Business Suite Mobile Release 10.x apps are compatible with Oracle E-Business Suite Release 12.2.3 and later. Oracle Field Service for EBS and Oracle Mobile SCM for EBS also support Oracle E-Business Suite Release 12.1.3.

Customization or Personalization of Oracle E-Business Suite Mobile Apps

For customization and personalization information for the Oracle E-Business Suite Mobile Release 10.x apps, refer to the following documents:

> **Note:** Oracle E-Business Suite Mobile Release 10.x apps have been

renamed to distinguish them from the app versions built with earlier technology. For more information about the app names used in different mobile releases, see *Oracle E-Business Suite Mobile Apps, Release 12.1 and 12.2 Documentation Index*, My Oracle Support Knowledge Document 1641772.1.

- Oracle Approvals for EBS: Customization or personalization is currently not available.

  Note: Oracle Approvals for EBS supports custom approval type and custom modifications in a seeded approval type. Refer to "Section 5.2: Adding Custom Workflows as Approval Types and Customizing a Seeded Approval Type (Optional)" in My Oracle Support Knowledge Document 1642423.1, *Oracle Mobile Approvals for Oracle E-Business Suite Release Notes*.

- Oracle Field Service for EBS: See My Oracle Support Knowledge Document 2274361.1, *Oracle Mobile Field Service Custom Tables Support* and My Oracle Support Knowledge Document 2274420.1, *Oracle Mobile Field Service Multiplatform Client Side Customization*.

- Oracle Learning for EBS: Customization or personalization is currently not available.

- Oracle Maintenance for EBS: See My Oracle Support Knowledge Document 1923702.1, *Oracle Mobile Maintenance for Oracle E-Business Suite Release Notes*.

- Oracle Mobile SCM for EBS (MSCA): See My Oracle Support Knowledge Document 469339.1, *Oracle WMS Personalization Framework*.

- Oracle Self-Service HR for EBS: Customization or personalization is currently not available.

- Oracle Timecards for EBS: Customization or personalization is currently not available.

Oracle E-Business Suite Mobile APIs

Oracle provides APIs corresponding to selected functionality in the Oracle E-Business Suite mobile apps. These APIs published through the Oracle Integration Repository are available for use in custom app development. For example, if you build custom apps using any mobile app development framework of your choice, you can make use of these APIs.

> **Note:** Oracle Integration Repository is a component in Oracle E-Business Suite Integrated SOA Gateway. You can search, view, and deploy desired APIs as REST services through the repository for use in custom app development. Refer to Administering Native Services, *Oracle E-Business Suite Integrated SOA Gateway Implementation Guide*.

Use the business entity "Mobile Optimized API" to search for all mobile app APIs in the repository, or search by the business entity for a functional area.

Before searching and using these APIs for your custom app development, ensure that you have applied server-side patches required for your app to have these APIs available in your Oracle E-Business Suite instance. For the server-side patch details, see the Setting Up the Mobile Apps chapter under Oracle E-Business Suite Mobile Apps Release 10.x in the *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

For a list of the mobile APIs published in the Oracle Integration Repository for each app, see Oracle E-Business Suite Mobile APIs Available in the Oracle Integration Repository, page A-1.

If you plan to deploy Oracle E-Business Suite mobile APIs as REST services for custom app development, then you must first configure Oracle E-Business Suite Integrated SOA Gateway for REST services. For information on configuring Oracle E-Business Suite Integrated SOA Gateway, refer to "Configuring Oracle E-Business Suite REST Services" section, *Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12.2*, My Oracle Support Knowledge Document 1311068.1. Ensure that you apply the latest patches for REST services in My Oracle Support Knowledge Document 1311068.1.

Part 2

Oracle E-Business Suite Mobile Apps Release 9.x and Earlier

2

Introduction to Oracle E-Business Suite Mobile Apps Release 9.x and Earlier

Overview

Mobile access to enterprise applications is fast becoming a standard part of corporate life. Such applications increase organizational efficiency because mobile devices are more readily at hand than their desktop counterparts.

There are a number of scenarios where the standard functionality delivered by Oracle E-Business Suite may not be sufficient for the standard desktop worker. In recent years, the business requirements have expanded beyond connecting from a single location through a single device. Meeting this need effectively requires specific mobile capabilities.

This document provides recommendations and guidance for creating mobile apps that are designed to integrate and work optimally with Oracle E-Business Suite Releases 12.1.3 and 12.2. The objective is to describe the different types of application, outline the certified technologies that can be used, list the recommended interfaces, and outline the security options that are available.

From the integration perspective, the Oracle E-Business Suite web service infrastructure does not depend on a specific client type. This means that service implementation can work seamlessly, regardless of the platform being used for the client applications (native, hybrid or web applications).

Oracle E-Business Suite recommends using Oracle Mobile Application Framework (MAF) to build smartphone apps. Oracle MAF is a hybrid mobile architecture, one that uses HTML5 and CSS to render the user interface, Java for the application business logic, and Apache Cordova to access device features, such as e-mail, contacts, camera, geolocation, and so on. Because Oracle MAF uses these cross-platform technologies, you can build an app that runs on both Android and iOS devices without having to use any platform-specific tools. After deploying a MAF application to a device, the application behaves similarly to applications that are created using platform-specific tools, such as Objective C or Android SDK. Furthermore, Oracle MAF enables you to

build the same application for smartphones or for tablets, thereby letting you reuse the business logic in the same application and target various types of devices, screen sizes, and capabilities. A MAF application installs on a user's device like any other application on the device.

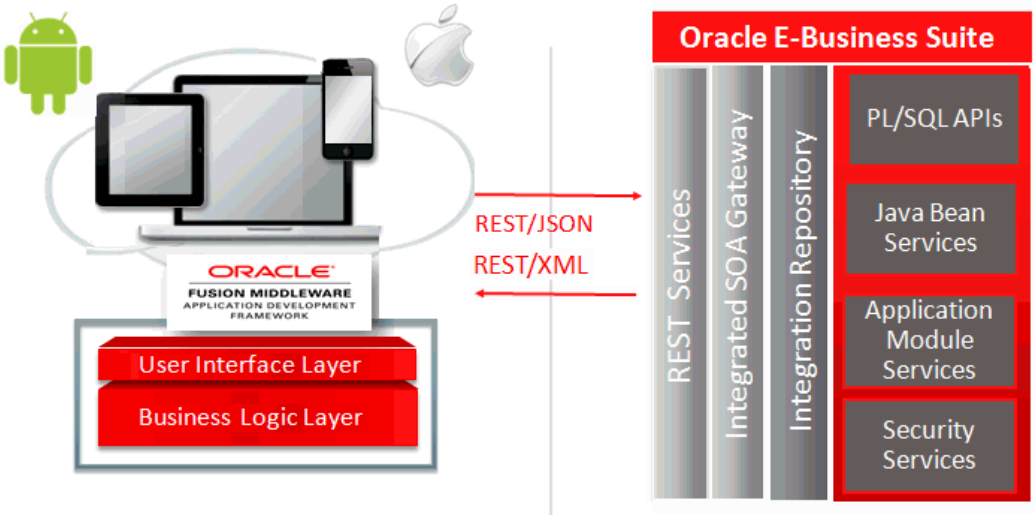
Oracle E-Business Suite Mobile Apps Developed Using Oracle Mobile Application Framework

For more information on the benefits of using Oracle MAF, refer to the Oracle Mobile Application Framework page available at `https://www.oracle.com/application-development/technologies/maf/oraclemaf.html`.

Technology Infrastructure

Oracle E-Business Suite mobile apps architecture consists of three technology layers: presentation, business logic, and data.

Oracle E-Business Suite Mobile Apps Technology Layers



- The **Presentation** layer consists of the user interface development framework (such as SDK and APIs), plus languages and platforms (such as Oracle MAF and HTML5) that determine how the end users will interact with an app.
- The server-side **Business Logic** layer represents application framework and programming languages that are used to develop programs that implement business process rules through Java or .NET based applications.
- The **Data** layer consists of programs that store and update data in the database that acts as the common repository.
- The Presentation layer communicates with the Business Logic layer, sending and receiving data. In turn, the server-side application communicates with the Data layer.

Presentation Layer

Oracle MAF lets you develop mobile apps using technologies such as HTML5, JavaScript, CSS, and Java, then deploy them to mobile device platforms, such as iOS and Android. Oracle MAF supports integration with native devices and also supports offline application interaction.

Business Logic Layer

Going by industry best practice, the core business logic that determines the functionality and behavior of an application should be reusable regardless of the presentation layer platform. Within Oracle E-Business Suite, the business logic resides in PL/SQL and Java-based Application Module Services.

> **Note:** Application Module Implementation class is a Java class that provides access to business logic governing the OA Framework-based components and pages. Such Java classes are called Application Module Services and are categorized as a subtype of Java interface.

The Oracle MAF-based client apps can interact with server-side logic through REST services. Oracle E-Business Suite REST services provided through Oracle E-Business Suite Integrated SOA Gateway (ISG) are available for Oracle E-Business Suite mobile apps. Oracle E-Business Suite public interfaces written in PL/SQL or Java-based Application Module Services can be deployed as REST services.

Please note that REST services are deployed in Oracle E-Business Suite's application server, and this type of web services does not depend on Oracle Fusion Middleware components like Oracle SOA Suite.

REST services are available in Oracle E-Business Suite Release 12.1.3 and Release 12.2.

Data Layer

All Oracle E-Business Suite transaction data resides in the Oracle database on which Oracle E-Business Suite is installed. An application's business logic either runs on the application tier Java-based interfaces or PL/SQL code that reads from the database tables. Oracle E-Business Suite data is retrieved by the business logic layer, based on an application-specific security context.

For more information on the integration options for Oracle E-Business Suite, refer to *Oracle E-Business Suite Release 12.1.3: Integration Products and Technologies Primer*, My Oracle Support Knowledge Document 1494997.1.

Oracle Mobile Application Framework Runtime Architecture

Oracle MAF is based on a hybrid mobile architecture and an extension of Apache Cordova (see `http://cordova.apache.org`). It enables a combination of HTML5 and ADF-defined pages and task flows to be rendered in the same downloadable application.

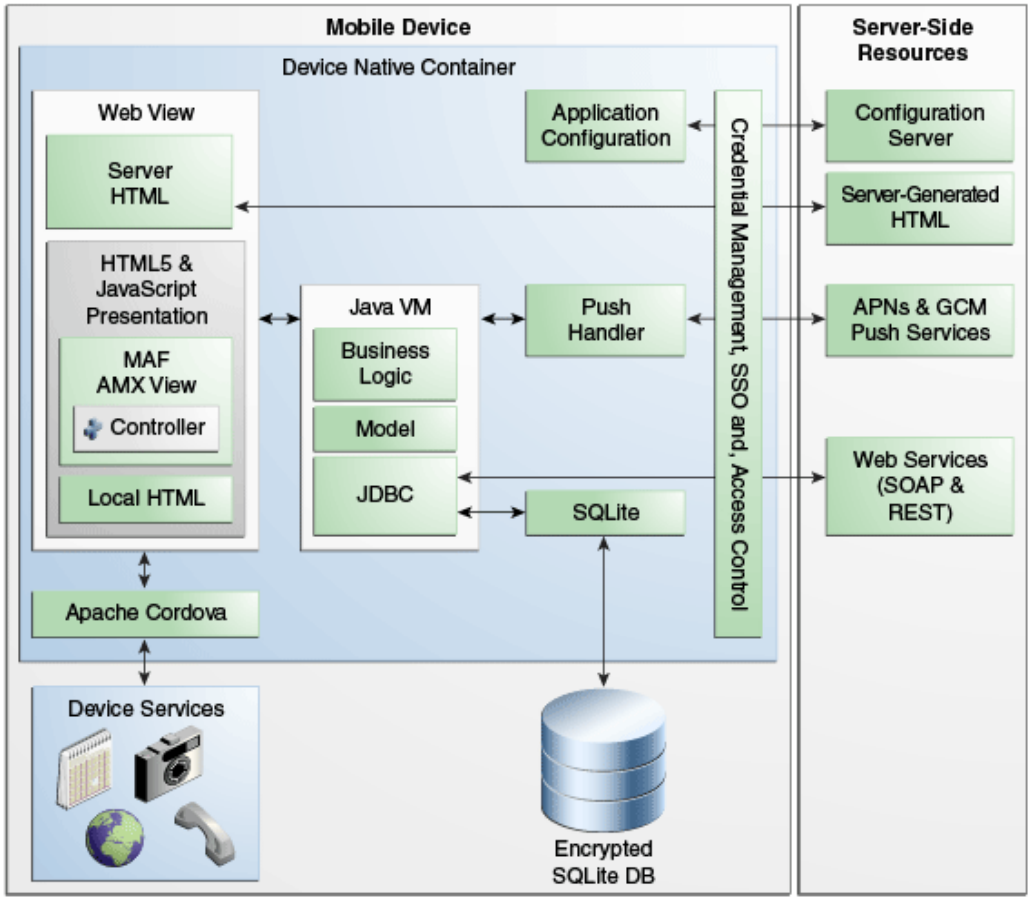
Oracle MAF consists of the following parts:

- Web View
- Cordova
- Java Virtual Machine
- Application Configuration
- Credential Management, SSO and Access Control

- Push Handler

Note: Oracle MAF's model-view-controller (MVC) stack resides on a mobile device and represents a reimplementation of ADF's model-view-controller layers. UI metadata is rendered to native components on the device, and bound to the model through the ADF model.

Oracle Mobile Application Framework Runtime Architecture



For more information about Oracle MAF, refer to *Developing Mobile Applications with Oracle Mobile Application Framework*.

User Experience

Mobile design patterns should be used when designing smartphone and other small format apps for platforms such as iPhone and Android. Design patterns are common flow or page designs that are intended for use across different product families. These patterns are built using industry best practices to meet common requirements, and

extensively tested for usability. A complete set of mobile design patterns has been defined by user experience, and can be re-created with the mobile components and technology delivered by ADF and Oracle JDeveloper.

Mobile design patterns can be used across mobile platforms and fully support designs for apps running in a mobile browser. Platform-specific designs and style sheets are not yet available. The finalized Fusion Mobile look and feel is not represented in the current design patterns, and will be provided when available.

For more information on User Experience, consult the Mobile Patterns page at https://www.oracle.com/webfolder/ux/mobile/patterns.html.

Accessibility

Accessibility involves making your app usable for people with disabilities such as low vision/blindness, deafness, or other physical limitations. This means creating apps that can be used without a mouse (keyboard only), used with a screen reader, and generally used without reliance on sound, color, or animation and timing.

Hybrid and native mobile apps must meet the Web Content Accessibility Guidelines (WCAG) 2.0 (`http://www.w3.org/TR/WCAG20/`) and Section 508 (`http://www.section508.gov/section-508-standards-guide`) standards. Oracle MAF provides accessibility guidelines for components, page, and navigation structures. Although useful, these are not a substitute for familiarity with accessibility standards and performing accessibility testing with assistive technology. Apple and Android provide detailed information on their accessibility support, including information on testing and making your app accessible.

For more information, consult the following documents:

- Web Content Accessibility Guidelines (WCAG) 2.0 (`http://www.w3.org/TR/WCAG20/`)
- Section 508 Standards (`http://www.section508.gov/section-508-standards-guide`)
- Accessibility Programming Guideline for iOS (`https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/iPhoneAccessibility/Introduction/Introduction.html#//apple_ref/doc/uid/TP40008785`)
- Accessibility Guidelines for Android (`http://developer.android.com/guide/topics/ui/accessibility/index.html`)
- Oracle Corporate Accessibility Policies (`https://www.oracle.com/us/corporate/accessibility/policies/index.html`)
- Understanding MAF Support for Accessibility, Creating the MAF AMX User Interface, *Developing Mobile Applications with Oracle Mobile Application Framework*

3

Setting Up the Development Environment

Overview

If you do not want to use standard Oracle E-Business Suite mobile apps installed directly from public app stores, Oracle lets you deploy mobile apps against Oracle E-Business Suite in the following different ways:

- **Enterprise distribution for Oracle E-Business Suite mobile apps**

  Starting from Oracle E-Business Suite Mobile Foundation Release 4.0, enterprises can distribute Oracle E-Business Suite mobile apps that are developed based on Oracle E-Business Suite Mobile Foundation internally through the use of Mobile Application Archive (MAA) files.

  > **Note:** This functionality provided through the use of MAA files is also available for selected mobile apps that are not developed based on Oracle E-Business Suite Mobile Foundation. See: Understanding Mobile Application Archive (MAA) Files, page 4-2.

  For each Oracle E-Business Suite mobile app available on the public app stores for a given release, a corresponding MAA file is provided that lets enterprises distribute mobile apps to internal users on an internal corporate location. This means that instead of downloading Oracle E-Business Suite mobile apps from a public app store, enterprise users can download the apps directly from an enterprise's own site.

  For information on using MAA files for enterprise distribution, see Using Mobile Application Archives for Enterprise Distribution, page 4-1.

  Note that starting from Oracle E-Business Suite Mobile Foundation 7.0, enterprises can optionally enable and send push notifications to the mobile devices of their enterprise users when using Oracle Mobile Cloud Service or using Oracle Mobile Hub from Release 9.0 when an appropriate patch is applied. For more information, refer to Implementing Push Notifications, page 6-1.

- **Custom mobile app development for Oracle E-Business Suite**

  Starting from Oracle E-Business Suite Mobile Foundation Release 4.1, enterprises can develop custom mobile apps to meet the enterprise needs. This feature enables enterprises to:

  - Extend Oracle E-Business Suite functionality with custom mobile apps that are not covered by the Oracle E-Business Suite mobile apps available in public stores.

  - Develop custom apps for customized Oracle E-Business Suite applications.

  Custom mobile app development typically involves the following three high-level steps:

  1. Design mobile apps

     This is an important step before you begin to implement mobile apps. Refer to Using the Sample App as a Reference, page B-1 for different design patterns. For Oracle mobile user experience (UX) standards and design guidelines, refer to https://www.oracle.com/webfolder/ux/middleware/alta/index.html.

  2. Deploy required REST APIs

     If you would like to develop custom Oracle E-Business Suite APIs for use in custom app development, you can publish them in the Oracle Integration Repository and deploy them as REST services.

     Starting from Oracle E-Business Suite Mobile Foundation Release 6.1, Oracle provides seeded REST APIs used in Oracle E-Business Suite mobile apps that can also be used to develop custom apps. Enterprises can deploy desired APIs that are available in Oracle Integration Repository as REST services for use in custom apps development. For information on searching and testing these seeded APIs, refer to Implementing Oracle E-Business Suite REST Services, page 7-4.

  3. Implement the custom mobile apps

     After deploying required APIs as REST services, you can develop your custom mobile apps using one of the following options:

     - The Login component from Oracle E-Business Suite Mobile Foundation Release 4.1 or later

       Starting from Oracle E-Business Suite Mobile Foundation Release 7.0, you can optionally enable push notifications when using Oracle Mobile Cloud Service or using Oracle Mobile Hub, from Release 9.0 when an appropriate patch is applied, for the custom apps developed using the Login component.

       For more details about implementing this feature for the custom mobile

apps, see: Implementing Push Notifications, page 6-1.

- Any mobile app development framework of your choice

Using Oracle E-Business Suite Mobile Foundation Login Component

The Oracle E-Business Suite Mobile Foundation Login component is an extension to Oracle Mobile Application Framework to rapidly develop and deploy custom mobile apps against Oracle E-Business Suite that provides required flows, such as configuration, login, responsibility selection, and so on.

For information on custom app development for Oracle E-Business Suite mobile apps, see Using the Login Component to Develop Mobile Apps, page 5-1.

> **Important:** Oracle Mobile Application Framework development knowledge is required for developing custom mobile apps for Oracle E-Business Suite using Oracle E-Business Suite Mobile Foundation. Refer to the Oracle Mobile Application Framework documentation (https://www.oracle.com/application-development/technologies/maf/oraclemaf.html).

This chapter describes the following topics on the environment setup for both enterprise distribution and custom app development for Oracle E-Business Suite:

- Performing Server-Side Tasks, page 3-3
- Performing Client-Side Tasks, page 3-23

Performing Server-Side Tasks

Perform the following tasks on the server side:

> **Note:** When setting up a development environment for Oracle Mobile Supply Chain Applications for Oracle E-Business Suite (MSCA), you only need to apply the Oracle E-Business Suite server-side patches described in this section. The mobile app definition metadata setup and migration between instances are not applicable for MSCA that is not developed based on Oracle E-Business Suite Mobile Foundation.

1. Applying Oracle E-Business Suite Server-Side Patches, page 3-4
2. Setting Up Mobile App Definition Metadata, page 3-4
3. Migrating Mobile App Metadata Between Instances, page 3-22

After applying the server-side patches and performing needed tasks, you need to

perform setup tasks on the client side. See: Performing Client-Side Tasks, page 3-23.

Applying Oracle E-Business Suite Server-Side Patches

Oracle strongly recommends applying the latest server-side patches for your mobile apps.

For the product family patch information, see Applying Prerequisite Patches on the Oracle E-Business Suite Server, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Setting Up Mobile App Definition Metadata

Oracle E-Business Suite Mobile Foundation based mobile apps can be configured and managed from Oracle E-Business Suite server.

Once an app is launched, it runs the validation steps against the Oracle E-Business Suite server to check if the app is configured and if the current user can access the app before allowing the user to use the app. Therefore, before deploying a mobile app that you have modified for enterprise distribution or created for custom app development, you need to perform the following setup tasks on the server to ensure the app is ready for deployment:

> **Note:** Since enterprises can develop custom mobile apps either using the Oracle E-Business Suite Mobile Foundation Login component or any mobile app development framework of their choices, the setup tasks described in this section are not required if the Login component is not used in custom app development.

> **Note:** For Oracle E-Business Suite mobile apps that you installed directly from a public app store, such as Apple App Store or Google Play, the definitions of these apps are already shipped by Oracle.

- Registering and Updating Your Mobile App Definition Metadata, page 3-5

  You must register your mobile app definition metadata first before deploying and testing the app against the Oracle E-Business Suite server.

- Creating and Using Mobile App Access Roles, page 3-19

  Oracle E-Business Suite mobile apps use role-based access control to secure mobile app data. To secure your mobile apps, you need to set up required app-specific mobile app access roles first so that administrators can assign these roles to responsibilities later. Users who have the mobile app access roles through those responsibilities will have access to the corresponding mobile apps.

- Configuring Your Mobile Apps, page 3-21

Prior to the deployment of your app, ensure that the app can work as expected. You need to work with administrators to configure and validate the app against an Oracle E-Business Suite environment.

Registering and Updating Your Mobile App Definition Metadata

You need to register your mobile app definition metadata on the server first before deploying and testing the app. If any changes to the mobile definition later, you can modify the application definition metadata if needed.

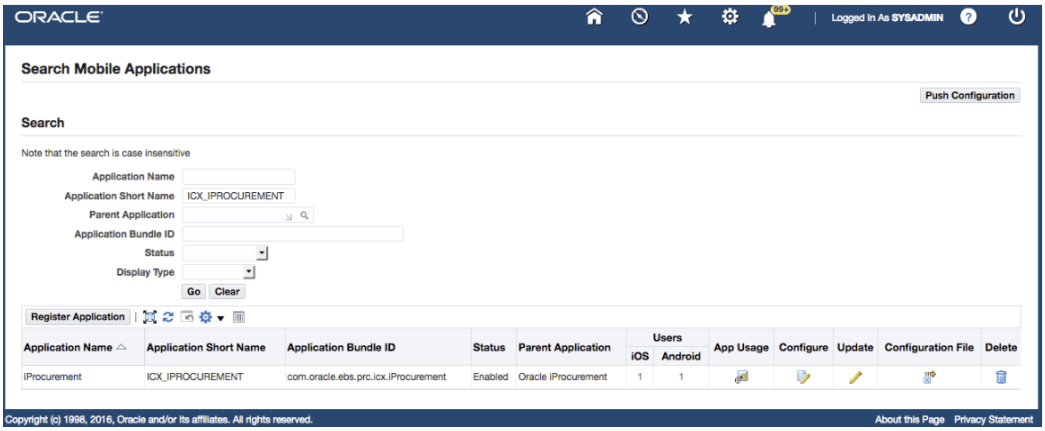
Accessing the Mobile Applications Manager UI Pages

To access the Mobile Applications Manager UI pages, log in to Oracle E-Business Suite as a user who has the Mobile Applications Manager responsibility.

> **Note: Separate User Privileges in the Mobile Applications Manager UI Pages**
>
> The Mobile Applications Manager responsibility allows you to access the Mobile Applications Manager UI pages, and this responsibility is assigned through the Mobile Applications Administrator role (UMX|FND_MBL_ROLE_ADMIN) or the Mobile Applications Developer role (UMX|FND_MBL_ROLE_DEV) which provides you with various access privileges. For more information about these user roles and their privileges, see Enabling a Mobile App Individually and Specifying the Configuration Through the UI Pages, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

The Search Mobile Applications page appears.

Search Mobile Applications Page



> **Note:** Users who have the Mobile Applications Administrator role can

find the additional Users and App Usage columns from the search result table for viewing the platform-specific user installation and app usage information. See: Viewing Mobile App Installation and Usage Metrics, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

For additional administrative tasks on enabling and configuring a mobile app individually through the UI pages or configuring multiple apps using a script as well as viewing configuration details, see the *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

This is the entry point where you can:

- Register an app by clicking the **Register Application** button. See: Registering Your Mobile App, page 3-6.
- Update the application definition metadata for an existing app by clicking the **Update** icon from a search result table. See: Updating Your Mobile App Definition, page 3-16.
- Delete an existing mobile app's application definition metadata by clicking the **Delete** icon from the search result table. See: Deleting Your Mobile App Definition, page 3-18.

Registering Your Mobile App

Use the following steps to register an app on the server:

1. Log in to Oracle E-Business Suite as a user who has the Mobile Applications Developer role.
2. Select the Mobile Applications Manager responsibility and choose the **Applications** link from the navigator.
3. In the Search Mobile Applications page, click the **Register Application** button to register an enterprise app.

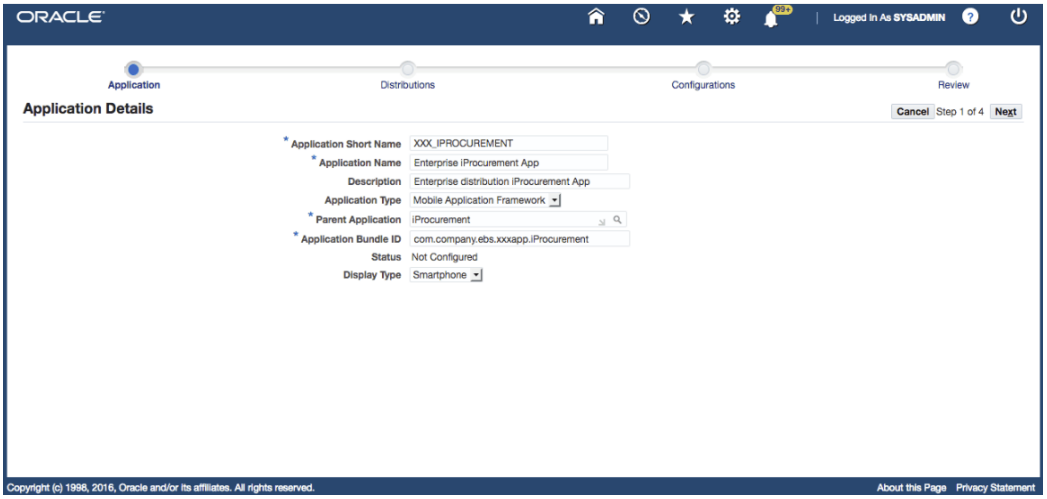
   The following pages are displayed in the sequence listed here as part of the registration process for an app:

   1. Application Details Page, page 3-7
   2. Distributions Page, page 3-8
   3. Configuration Details Page, page 3-9
   4. Review Page, page 3-15

Application Details Page

After you click the **Register Application** button in the Search Mobile Applications page to register an app, the Application Details page appears.

Application Details Page



Enter the following application metadata information for your enterprise app:

- **Application Short Name:** Enter the short name for the mobile app, such as "XXX_IPROCUREMENT".
- **Application Name:** Enter the display name for the mobile app.
- **Description:** Enter optional description information for the mobile app
- **Application Type:** "Mobile Application Framework" is selected by default.
- **Parent Application Name:** Enter the Oracle E-Business Suite application to which the mobile app belongs, such as "iProcurement".
- **Application Bundle Id:** Enter a unique bundle identifier for the mobile app, such as `com.company.ebs.xxxapp.iProcurement` for your mobile app.

  This value will be used later in the Id field of the `maf-application.xml` file in Oracle JDeveloper for the mobile app.
- **Display Type:** "Smartphone" is selected by default.
- **Status:** "Not Configured" is selected by default.

  When the app is configured, the status should be set to "Enabled". For information on configuring your mobile apps, see Configuring the Mobile Apps on the Oracle E-Business Suite Server, Setting Up the Mobile Apps chapter, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Note: Ensure that the desired information for Application Name, Application Short Name, and Application Bundle Id are correctly specified for your app. These fields are not allowed to change after the registration. See: Updating Your Mobile App, page 3-16.

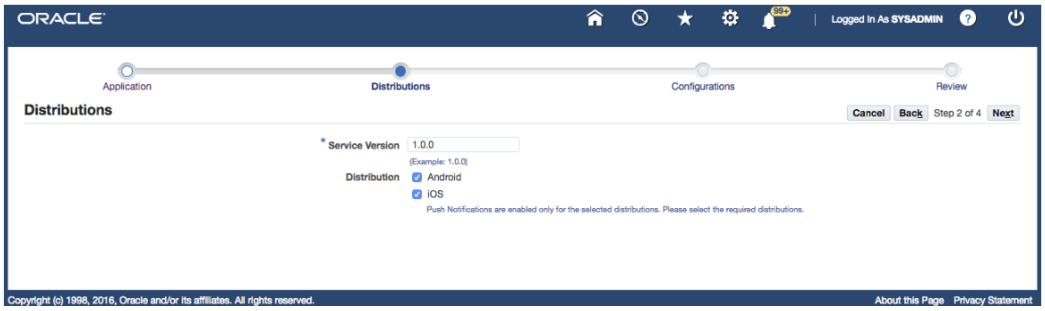
Click **Next** to access the Distributions page to continue the registration process. See: Distributions Page, page 3-8.

For information on registering an app, see Registering Your Mobile App, page 3-6.

Distributions Page

After you enter the application metadata information for an app and click **Next**, the Distributions page appears. This is the page where you specify the service version and distribution method for the app.

Distributions Page



Enter the following distribution information for the app:

- **Service Version:** Enter the service version number corresponding to the app's REST service implementation.
  - Enterprise distribution for Oracle E-Business Suite mobile apps

    This service version number must be exactly the same as the service version of the corresponding Oracle E-Business Suite mobile app. This is because the app uses the same REST services as that of the corresponding seeded mobile app. This service version number is important for mobile clients to access and determine the server's capability. Therefore, this service version number cannot be different from the corresponding seeded app.

    For example, if the service version of the seeded Oracle Mobile iProcurement for Oracle E-Business Suite app is "1.0.0", you must enter the same service version "1.0.0" here.

    To obtain the service version information of the corresponding seeded app, locate the seeded app through a search first, and then click the app's Application Name link from the search result table. The service version information is displayed in the read-only Application Details page.

If you upgrade an enterprise-distributed app to a new version of the corresponding MAA file and apply the corresponding consolidated product family patches, ensure to manually check and update the Service Version number of the enterprise app to that of the value in the corresponding seeded app. For information on initiating the server updates, see Directing Users to Obtain Connection Details and Initiate Server Updates, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

- Custom mobile apps

  For custom apps, use this Service Version field to maintain the version number for the app's REST service implementation. This version number could start with 1.0.0 and increment only when the app's REST service implementation is enhanced significantly with new features. The Service Version value is available to the mobile app code through application scope variable `{applicationScope.EBSServiceVersion)`. You can use the value on the mobile app to determine the server capability.

- **Distribution:** Select the desired distribution platform check box(es) for your app, such as Android, iOS, or both platforms. By default, both the Android and iOS check boxes are selected.

  If you plan to enable push notifications for both platforms, select both check boxes for iOS and Android. Otherwise, push notifications will not work for the unselected platform.

Click **Next** to access the Configurations page. See: Configuration Details Page, page 3-9.

Click **Back** to modify the application metadata information if needed. See: Application Details Page, page 3-7.

For information on registering an enterprise app, see Registering Your Mobile App, page 3-6.

Configuration Details

Use the Configuration Details page to specify the desired configuration parameters for your mobile apps. Specifically, you can use this page to:

- Configure a desired authentication type and set the related parameters

  See: Configuring a Desired Authentication Type, page 3-9.

- Configure push notifications with desired parameters for the supported mobile apps (available from Oracle E-Business Suite Mobile Foundation Release 7.0 and onwards)

  See: Adding Push Notifications to App Configuration, page 3-13.

Configuring a Desired Authentication Type

Selecting "Connection Settings" as the Category Value for Configuring a Desired

Authentication Type

Category "Connection Settings" is required to be configured for every mobile app for it to connect to the Oracle E-Business Suite server. You must select either one of the following authentication types for your app based on your authentication scenarios as the Sub Category value for the connection settings:

- Apps Local Login (previously known as "HTTP Basic")

  This type corresponds to the "HTTP Basic" authentication server type used in Oracle Mobile Application Framework.

- Apps SSO Login (previously known as "Web SSO")

  This type corresponds to the "Web SSO" authentication server type used in Oracle Mobile Application Framework.

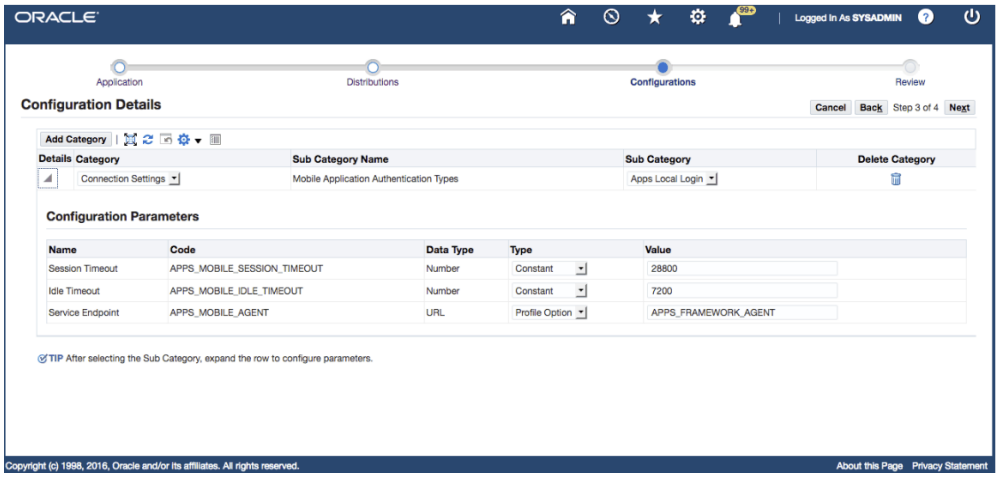
  Note: The authentication type selected here during the app registration can be changed later when an administrator configures the app.

- **Apps Local Login (default) - for local authentication**

  This is the default authentication type for the connection settings where users are authenticated locally against the Oracle E-Business Suite server. The corresponding configuration parameters for the Apps Local Login type are:

  - APPS_MOBILE_SESSION_TIMEOUT
  - APPS_MOBILE_IDLE_TIMEOUT
  - APPS_MOBILE_AGENT

Configuration Details Page with Configuration Parameters for Apps Local Login



For information on setting configuration parameters for the Apps Local Login authentication type, see step 8 in the Enabling and Configuring a Mobile App Individually section, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

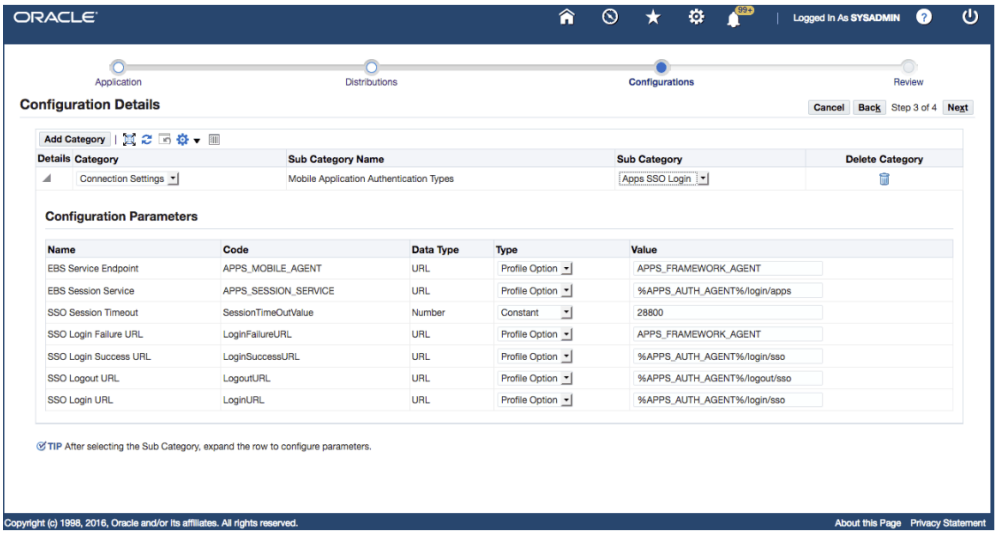
- **Apps SSO Login - for remote authentication**

  Use the Apps SSO Login authentication type if you want to delegate authentication to Oracle Access Manager.

  Before selecting this authentication type, ensure that your Oracle E-Business Suite instance is integrated with Oracle Access Manager. Refer to My Oracle Support Knowledge Document 1388152.1, *Overview of Single Sign-On Integration Options for Oracle E-Business Suite*. You must apply required patches and perform additional setup tasks to enable this feature. See: Mobile Specific Setup Tasks to Enable Apps SSO Login Authentication Security, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

  For more information about single sign-on, see Advanced Configurations for Single Sign-On (SSO), *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Configuration Details Page with Configuration Parameters for Apps SSO Login



Once "Apps SSO Login" is selected as the authentication type, the following corresponding configuration parameters are automatically displayed in the Configuration Parameters region:

- APPS_MOBILE_AGENT
- APPS_SESSION_SERVICE
- SessionTimeOutValue
- LoginFailureURL
- LoginSuccessURL
- LogoutURL
- LoginURL

For information on setting configuration parameters for the Apps SSO Login authentication type, see step 8 in the Enabling and Configuring a Mobile App Individually section, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Configuration Parameters region

The Configuration Parameters region allows you to predefine the configuration parameters for a selected configuration category. To display this region, click the **Show** link next to the configuration category.

Specify the parameter values listed in the region to predefine the configuration settings for the app.

- **Name:** This is the configuration parameter name corresponding to the category and sub-category specified earlier in the Configuration Categories region.

- **Type:** This determines how the default value for a parameter is assigned. Select either one of the following values:

  - Constant - It indicates that the value for the parameter is a constant.

    For example, if "Constant" is selected, enter a constant value "28800" seconds as the default value in the Value field for the Session Timeout (APPS_MOBILE_SESSION_TIMEOUT) parameter.

  - Profile Option - It indicates a profile option is used to retrieve the value for the parameter.

    If this is selected, enter the profile option name in the Value field for the parameter. For example, enter "APPS_FRAMEWORK_AGENT" as the default profile option name in the Value field for the Service Endpoint (APPS_MOBILE_AGENT) parameter.

- **Value:** This is the default value for the parameter. This value is either a valid profile option internal name or a constant value depending on the value selected in the Type field.

  For example, the default value for the SSO Login URL (LoginURL) parameter is the current value of `%APPS_AUTH_AGENT%/login/sso`.

  Please note that the convention `%<string>%` is used specifically for parameter values of type "Profile Option" and the value of which contains content that is in addition to the profile value. For example, the runtime value of this SSO Login URL parameter would be `<profile-value-of-the-APPS_AUTH_AGENT>/login/sso`, where `/login/sso` is a constant.

Click **Next** to access the Review page. See: Review Page, page 3-15.

Click **Back** to modify the configuration information if needed. See: Distributions Page, page 3-8.

For information on registering an app, see Registering Your Mobile App, page 3-6.

Adding Push Notifications to App Configuration

Starting from Oracle E-Business Suite Mobile Foundation 7.0 and onwards, Oracle E-Business Suite supports push notifications in the following apps, when using Oracle Mobile Cloud Service or using Oracle Mobile Hub (from Release 9.0 and onwards when an appropriate patch is applied):

- Custom Oracle E-Business Suite mobile apps developed using the Login component from Oracle E-Business Suite Mobile Foundation

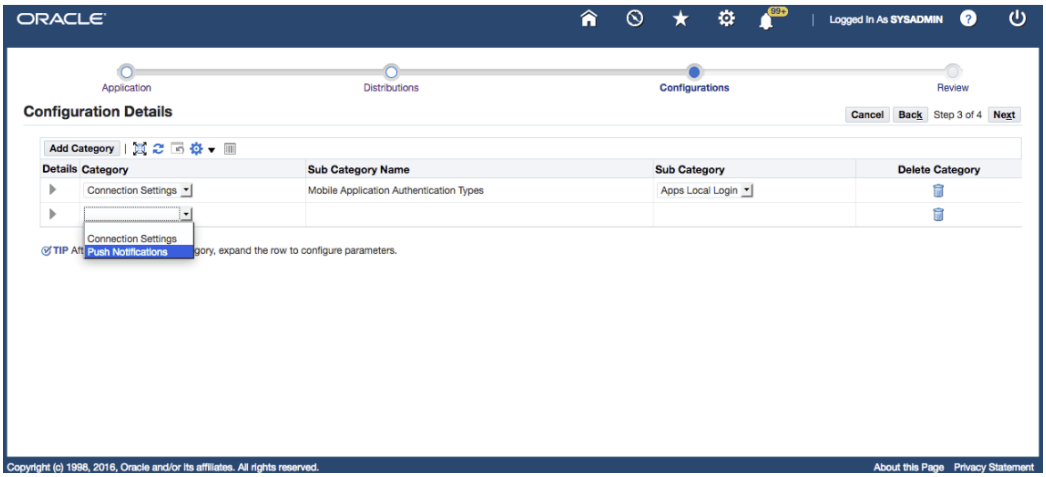
- Oracle Mobile Approvals for Oracle E-Business Suite, when provided to users through enterprise distribution

Selecting "Push Notifications" as the Category Value for Configuring Push Notifications

If your mobile app supports push notifications, in order to enable the feature and configure the related parameters, you must add the "Push Notifications" category.

Click the **Add Category** button to add another row. Select "Push Notifications" as the Category value. "No" is displayed as the default value in the Sub Category field which means that when the app is deployed, the push notifications feature is disabled by default.

Configuration Details Page with "Push Notifications" Category Selected



In addition to adding the "Push Notifications" category for the mobile app definition and configuring corresponding parameters, following tasks are important for the server and the mobile app to send and process push notifications:

- Perform required changes for the mobile apps.

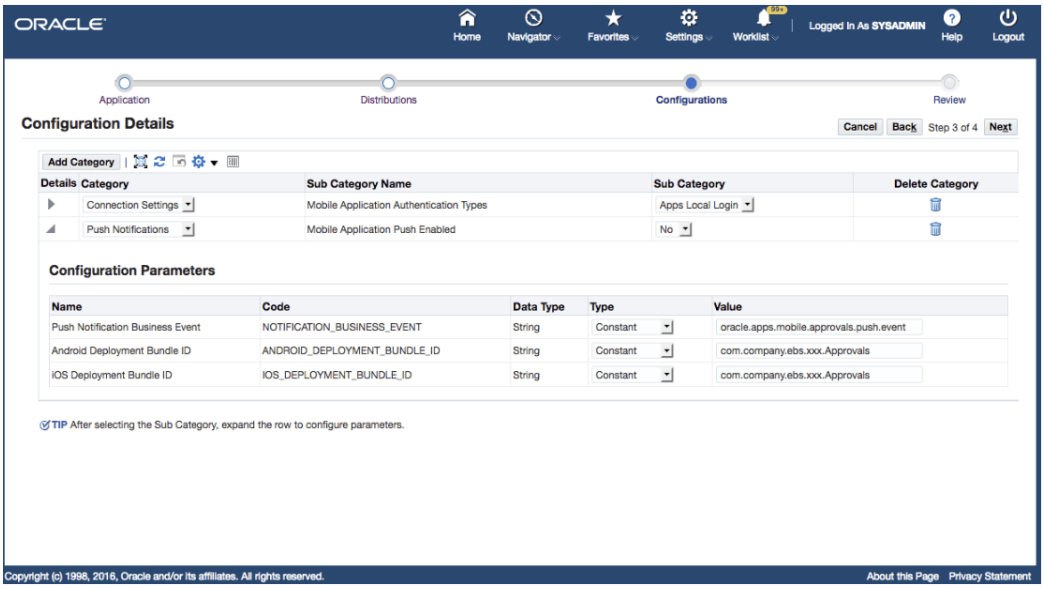
  See: Implementing Push Notifications, page 6-1.

- Perform required tasks on the Oracle E-Business Suite server.

  See: Setting Up and Enabling Push Notifications for Oracle E-Business Suite Mobile Apps, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

In the Configuration Parameters region, specify default parameter values for the mobile app. The default parameter values can be overwritten by the administrator when configuring the mobile app.

Configuration Details Page with Configuration Parameters for Push Notifications



> **Note:** You must specify the Push Notifications Business Event field, but you may leave the Android Deployment Bundle ID and iOS Deployment Bundle ID fields blank when creating the mobile app definition metadata.

- Push Notifications Business Event (NOTIFICATION_BUSINESS_EVENT): The app-specific business event used to trigger push notifications from the Oracle E-Business Suite application code when the event arises.

  For example, Oracle Workflow server code uses `oracle.apps.mobile.approvals.push.event` business event to send push notifications to the enterprise-distributed Approvals app.

- Android Deployment Bundle ID (ANDROID_DEPLOYMENT_BUNDLE_ID): The Package Name used in the deployment profile to create the Android app.

- iOS Deployment Bundle ID (IOS_DEPLOYMENT_BUNDLE_ID): The Bundle ID registered with Apple to receive the iOS provisioning profile. This is the bundle ID used in the deployment profile to create the iOS app.

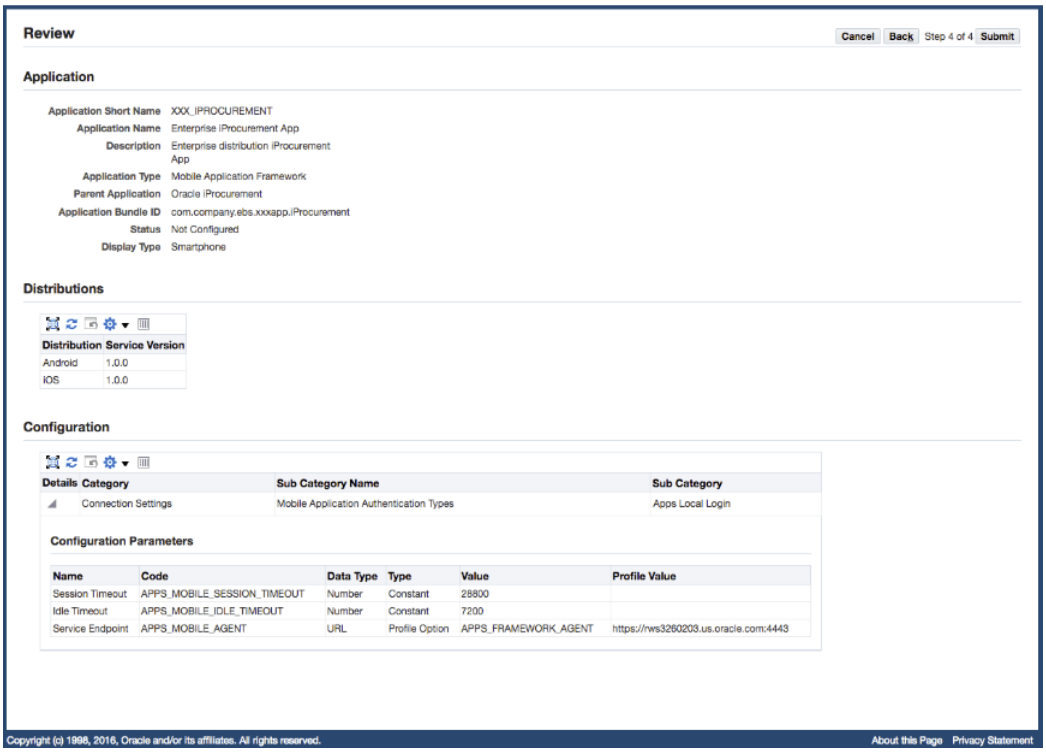
Click **Next** to access the Review Page, page 3-15.

For information on registering an app, see Registering Your Mobile App, page 3-6.

Review Page

Once you complete the required registration information for your app, the read-only Review page appears. It displays all the metadata information, distributions, and configuration details that you specified earlier for the app.

Mobile App Registration Information Displayed in the Review Page



If no more change is required for the app, click **Submit** to save and register the app. A confirmation message appears indicating that the mobile app is successfully registered.

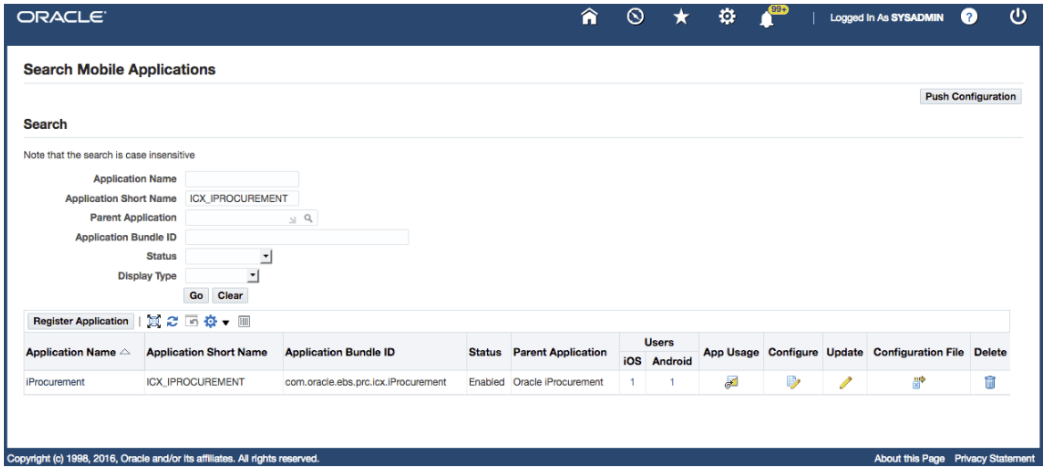
Click **Back** to modify the information if needed. See: Configuration Details Page, page 3-9.

For information on registering a mobile app, see Registering Your Mobile App, page 3-6 .

Updating Your Mobile App Definition

To update the definition of an existing app, locate the app from the search result table in the Search Mobile Applications page.

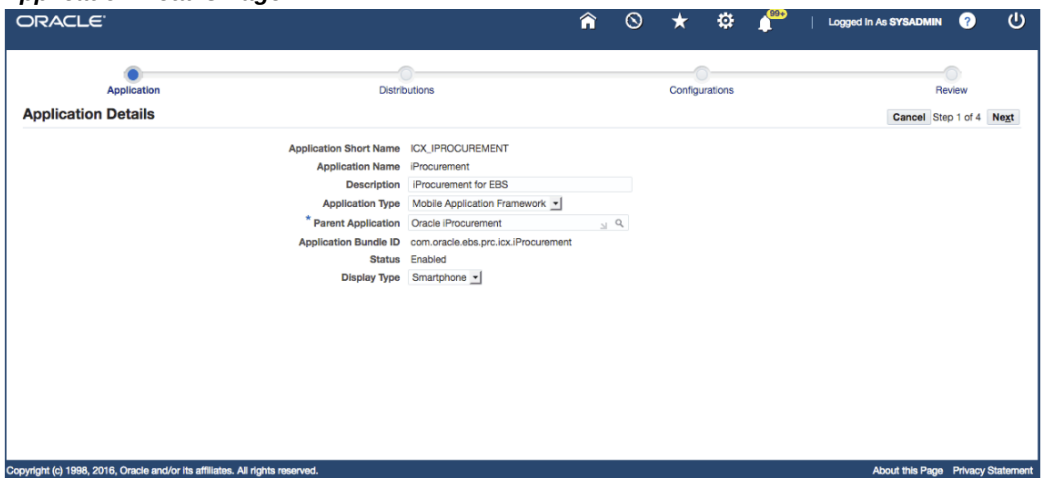
Search Mobile Applications Page



Click the **Update** icon from the search result table. This action allows you to access the Application Details page and modify the definition of the selected enterprise app.

> **Warning:** You can update the application definition metadata for an app that you register for enterprise distribution or custom app development, but should never remove or modify the application definitions of Oracle E-Business Suite mobile apps, starting with `com.oracle.ebs*`.

Application Details Page



> **Note:** Alternatively, you can click the Application Name link, such as ICX_IPROCUREMENT, from the search result table first to display the

read-only Application Details page. Click the **Update** button in the page to enable the update for the app.

Similar to the app registration process, you can update the application definition metadata for the selected app.

- **Updating the application definition metadata in the Application Details Page**

  In the Application Details page, you can update the application metadata information including description, application type, parent application, and display type.

  Please note that the Application Name, Application Short Name, and Application Bundle ID fields are not enabled for update.

- **Updating the distribution information in the Distributions Page**

  In the Distributions page, you can update the service version, and distribution methods for the selected app.

- **Updating the configuration information in the Configuration Details Page**

  In the Configuration Details page, you can change the Sub Category values for currently used configuration categories and corresponding parameter values, as well as add or remove configuration categories.

  Note that the "Connection Settings" category is required for all Oracle E-Business Suite mobile apps. See: Configuring a Desired Authentication Type, page 3-9.

  As mentioned earlier during the app registration, if "Apps SSO Login" is selected as the mobile app authentication type for an app, ensure that the Oracle E-Business Suite instance must be integrated with Oracle Access Manager. Refer to My Oracle Support Knowledge Document 1388152.1, *Overview of Single Sign-On Integration Options for Oracle E-Business Suite*. You must apply required patches and perform additional setup tasks to enable this feature. See: Mobile Specific Setup Tasks to Enable Apps SSO Login Authentication Security, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

For information on registering an app, see Registering Your Mobile App, page 3-5.

Deleting Your Mobile App Definition

When the application definition of an app becomes invalid or is no longer needed, you can remove it from the server.

> **Warning:** You can delete or update the application definition metadata for an app that you register for enterprise distribution or custom app development, but should never remove or modify the seeded application definitions of Oracle E-Business Suite mobile apps, starting

with `com.oracle.ebs*`.

To delete the definition of an app, click the **Delete** icon from the search result table. A confirmation message appears requiring you to confirm the delete action. Once it's confirmed, the definition of the selected app is removed from the database and it is no longer available for users.

Creating and Using Mobile App Access Roles

Oracle E-Business Suite mobile apps use role-based access control to allow users who are assigned the appropriate access roles to access Oracle E-Business Suite.

Similar to self-service applications, some mobile apps may require a responsibility to be selected before accessing required application data. In order to show a list of responsibilities for users to select before using a mobile app, you need to first create a mobile app access role, and then associate the access role with your mobile app in Oracle JDeveloper.

When the mobile app is deployed to users, a mobile applications administrator can then assign the mobile app access role to responsibilities that will have access to the mobile app. All users who have access to those responsibilities will then inherit that mobile app access role and hence be able to use one of those responsibilities before using the mobile app to access Oracle E-Business Suite.

> **Important:** Oracle Mobile Approvals for Oracle E-Business Suite is the only app that does not have an app-specific access role required for users to access the app. Other than the Approvals app, you must set up required roles for your apps.

1. Create mobile app access roles in Oracle E-Business Suite.

   See: Creating Mobile App Access Roles, page 3-19.

2. Assign the mobile app access roles to responsibilities.

   See: Assigning Mobile App Access Roles to Responsibilities, page 3-21.

3. Migrate the mobile app access role definitions to a target Oracle E-Business Suite instance where your app could connect.

   See: Downloading and Uploading Mobile App Access Roles, page 3-22.

Creating Mobile App Access Roles

Perform the following steps to create mobile app access roles:

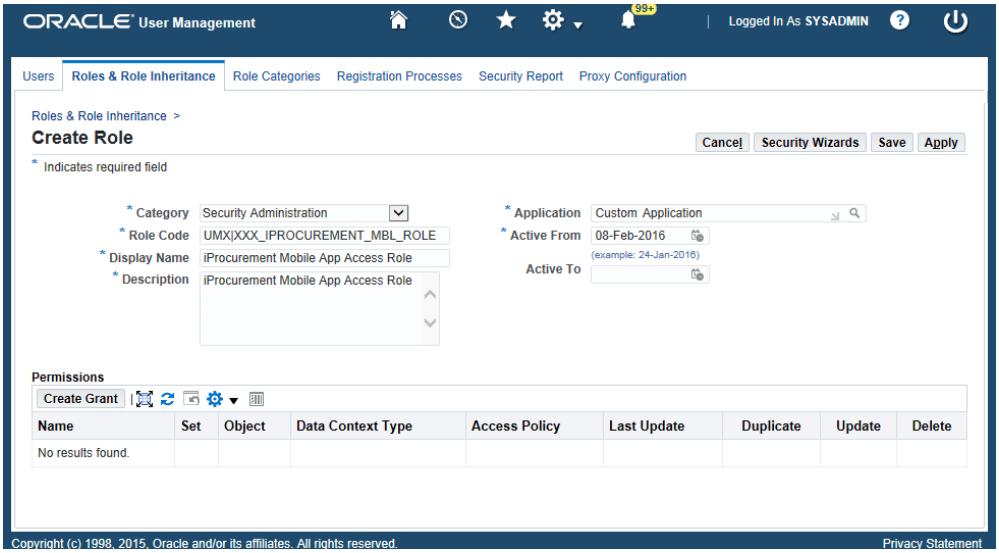
1. Log in to Oracle E-Business Suite as a user who has the User Management responsibility. For example, log in as `SYSADMIN`.

2. Select the User Management responsibility and navigate to the Roles and Role Inheritance page.

3. In the Roles and Role Inheritance page, click the **Create Role** button.

4. The Create Role page appears.

Create Role Page



Enter the following information in the Create Role page:

- **Category:** Select "Security Administration" from the drop-down list.

- **Role Code:** Enter the role code in the format of "`PROD_MBL_APP_NAME`", such as "`XXX_IPROCUREMENT_MBL_ROLE`".

  Record the Role Code information entered here which will be used later for the client-side setup.

  - A prefix "`UMX|`" is added to this value automatically.

    In this example, the entered code value is automatically converted to `UMX|XXX_IPROCUREMENT_MBL_ROLE`.

  - The Role Code information entered here should be specified in the `oracle.ebs.login.rolecode` property of the `ebs.properties` file later. See Assigning Mobile App Access Roles, page 4-21.

- **Display Name:** Enter a valid display name, such as "iProcurement Mobile App Access Role".

- **Description:** Enter a valid description information for the role, such as

"iProcurement Mobile App Access Role".

- **Application:** Select the application name. For example, "Custom Application".

  Similar to the Role Code value, the Application Short Name information entered here should be set up in the `oracle.ebs.login.roleappname` property of the `ebs.properties` file later. See Assigning Mobile App Access Roles, page 4-21.

- **Active From:** Leave the default unchanged.

- **Active To:** Leave this field blank.

5. Save your work.

Note that the REST interfaces created for a given mobile app should be granted to the mobile app access role created for that app. See: Implementing Oracle E-Business Suite REST Services, page 7-4.

Additionally, once a new mobile app access role is created for an enterprise app, make sure that the corresponding REST services permission set of the associated seeded app is granted to the new access role. For information on the seeded app-specific access roles and REST services permission sets for Oracle E-Business Suite mobile apps, see Appendix C: Mobile App Access Roles, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Assigning Mobile App Access Roles to Responsibilities

After you create app-specific access roles, a mobile applications administrator can then assign these roles to responsibilities.

If the mobile app access roles were not assigned to any valid responsibilities that are assigned to the mobile users, those users will not be able to see any responsibilities on the mobile app and hence will not be able to connect the mobile app to Oracle E-Business Suite.

For information on assigning roles to responsibilities, see Setting Up Mobile App Access to Responsibilities, Setting Up the Mobile Apps chapter, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Please note that on the client side you need to specify these app-specific roles in the MAF application's `ebs.properties` file in order for the corresponding apps to use these roles. See: Using Mobile App Access Roles, page 4-21.

Configuring Your Mobile Apps

Before deploying your apps, you must work with a mobile applications administrator to perform needed administrative tasks to ensure your apps can work as expected. These administrative tasks, including mobile app configuration and validation, can be performed in a development instance.

For information on the administrative tasks, see the *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Migrating Mobile App Metadata Between Instances

If there is a need to migrate the mobile application definition metadata and relevant mobile app access roles from one instance to another, perform the following tasks to transport the needed information:

This section includes the following topics:

- Downloading and Uploading Mobile App Definitions, page 3-22

  You can download the app definition metadata into a data (.ldt) file and then upload the file to another instance if needed.

- Downloading and Uploading Mobile App Access Roles, page 3-22

  Similar to the migration of the app definition metadata, this allows you to migrate mobile app access roles between instances.

Downloading and Uploading Mobile App Definitions

Once the application definition metadata has been registered in the database, you can transport the metadata information between different instances for testing or migration purposes.

Downloading Mobile App Definition Metadata

This can be achieved by first downloading the metadata into a data (.ldt) file, based on a `.lct` file `$FND_TOP/patch/115/import/afmobile.lct` using the Application Short Name as the key.

For example, use the following commands to download the metadata to a `.ldt` file:

```
FNDLOAD <APPS username> 0 Y DOWNLOAD
$FND_TOP/patch/115/import/afmobile.lct xxxiproc.ldt
FND_MBL_APPLICATION APPLICATION_SHORT_NAME=XXX_IPROCUREMENT

ORACLE Password:
```

Uploading Mobile App Definition Metadata

To upload the downloaded `.ldt` file, such as `xxxiproc.ldt`, to another instance, use the following commands:

```
FNDLOAD <APPS username> 0 Y UPLOAD
$FND_TOP/patch/115/import/afmobile.lct xxxiproc.ldt

ORACLE Password:
```

Downloading and Uploading Mobile App Access Roles

Similar to the concepts of transporting application definition metadata information between instances, you can migrate the mobile app access role definitions from one

instance to another if required.

Downloading Mobile App Access Roles

Use the following commands to download the definition of mobile app access roles:

```
FNDLOAD <APPS username> 0 Y DOWNLOAD
$FND_TOP/patch/115/import/afrole.lct xxxapprole.ldt WF_ROLE
ROLE_NAME=UMX|XXX_IPROCUREMENT_MBL_ROLE

ORACLE Password:
```

Uploading Mobile App Access Roles

After downloading the definition of mobile app access roles, you can upload the downloaded `.ldt` file to another Oracle E-Business Suite instance to which the mobile app should connect. For example, use the following commands to upload the role definition:

```
FNDLOAD <APPS username> 0 Y UPLOAD
$FND_TOP/patch/115/import/afrole.lct xxxapprole.ldt

ORACLE Password:
```

Performing Client-Side Tasks

This section describes the following setup tasks:

1. Installing the Development Tools, page 3-23
2. Downloading and Installing Oracle JDeveloper 12.2.1.3.0 Studio Edition, page 3-24
3. Downloading and Installing Oracle Mobile Application Framework 2.6.3 for Oracle E-Business Suite Mobile Foundation Release 9.1, page 3-24
4. Setting Up Oracle JDeveloper, page 3-26

Installing the Development Tools

Install the required development tools for the iOS and Android platforms:

- For the iOS platform, register with iOS Developer Program. Download and install Apple Xcode 10 and iOS SDK.
    - For mobile apps built with Oracle E-Business Suite Mobile Foundation Release 8.0, download and install Apple Xcode 9 and iOS SDK.
    - For mobile apps built with Oracle E-Business Suite Mobile Foundation Release 7.0, download and install Apple Xcode 8 and iOS SDK.

- For the Android platform, set up the downloaded Android SDK.

Downloading and Installing Oracle JDeveloper 12.2.1.3.0 Studio Edition

To develop mobile apps for Oracle E-Business Suite Mobile Foundation 9.1, 9.0, and 8.0, ensure you have Oracle JDeveloper version 12.2.1.3.0 Studio Edition.

> **Note:** For mobile apps built with Oracle E-Business Suite Mobile Foundation 7.0, ensure that you have Oracle JDeveloper version 12.2.1.0.0 Studio Edition.

For information on setting up Oracle JDeveloper, see *Oracle Fusion Middleware Installing Oracle JDeveloper*.

Downloading and Installing Oracle Mobile Application Framework 2.6.3 for Oracle E-Business Suite Mobile Foundation Release 9.1

Downloading Oracle Mobile Application Framework

In addition to Oracle JDeveloper, you need to download Oracle Mobile Application Framework 2.6.3 included in the "Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 9.1" software distribution from the Oracle Software Delivery Cloud (https://edelivery.oracle.com) through part number V1031711-01. You can also download this MAF version on My Oracle Support through Patch 31581902.

- If your MAA files are developed based on Oracle E-Business Suite Mobile Foundation Release 9.0, download "Oracle Mobile Application Framework 2.6.2 for Oracle E-Business Suite Mobile Foundation Release 9.0" (with additional bug fixes on top of Oracle MAF 2.6.2) included in the "Oracle E-Business Suite Mobile Application Archive 9.0" software distribution instead from the Oracle Software Delivery Cloud (https://edelivery.oracle.com) through part number V995061-01. You can also download this MAF version on My Oracle Support through Patch 30707039.

- If your MAA files are developed based on Oracle E-Business Suite Mobile Foundation Release 8.0, download "Oracle Mobile Application Framework 2.5.0 for Oracle E-Business Suite Mobile Foundation Release 8.0" (with additional bug fixes on top of Oracle MAF 2.5.0) included in the "Oracle E-Business Suite Mobile Application Archive 8.0" software distribution instead from the Oracle Software Delivery Cloud (https://edelivery.oracle.com) through part number V978633-01. You can also download this MAF version on My Oracle Support through Patch 27983689.

- If your MAA files are developed based on Oracle E-Business Suite Mobile Foundation Release 7.0, download "Oracle Mobile Application Framework 2.4.0 for Oracle E-Business Suite Mobile Foundation Release 7.0" (with additional bug fixes on top of Oracle MAF 2.4.0) included in the "Oracle E-Business Suite Mobile Application Archive 7.0" software distribution instead from the Oracle Software Delivery Cloud (https://edelivery.oracle.com) through part number V861706-01.

- You can also download this specific Oracle MAF version at https://www.oracle.com/application-development/technologies/maf/mafdownload.html.

Additionally, install underlying mobile platforms required for Oracle Mobile Application Framework. For example, in Oracle E-Business Suite Mobile Foundation Release 9.1, install the following platforms required for Oracle Mobile Application Framework 2.6.3:

- iOS - Xcode 11.x

- Android - Compile API Level 29

For information on maintaining two instances, refer to Oracle Mobile Application Framework 2.6.3 Migration Notes (from MAF 2.6.2) (https://www.oracle.com/application-development/technologies/maf/maf263migration.html).

- For Oracle E-Business Suite Mobile Foundation Release 9.0, see Oracle Mobile Application Framework 2.6.2 Migration Notes (https://www.oracle.com/application-development/technologies/maf/maf262migration.html).

- For Oracle E-Business Suite Mobile Foundation Release 8.0 and Release 7.0, refer to "Using Xcode 9.x with MAF 2.5.0" and "Using Xcode 8 and Deploying to iOS 10 with MAF 2.4.0" respectively in *Installing Oracle Mobile Application Framework*.

Installing Oracle Mobile Application Framework

After you have installed the iOS SDK, Android SDK, or both required for your platform (s) and downloaded the required Oracle Mobile Application Framework 2.6.3, follow the installation instructions to install the downloaded Oracle Mobile Application Framework. See: Installing the MAF Extension in JDeveloper, Installing Mobile Application Framework with JDeveloper, *Installing Oracle Mobile Application Framework*.

You can use Oracle Mobile Application Framework for:

- Working with Mobile Application Archive files to create enterprise-distributed Oracle E-Business Suite mobile apps

  For information on downloading Mobile Application Archive files, see:

Downloading Mobile Application Archives Files, page 4-5.

- Developing custom apps for Oracle E-Business Suite using Oracle E-Business Suite Mobile Foundation Login Component

  See: Using the Login Component to Develop Mobile Apps, page 5-1.

Setting Up Oracle JDeveloper

To configure the environment for your target platform in Oracle JDeveloper. select **Tools** from the menu and then **Preferences** to open the Preferences dialog. Select your desired platform, either Android or iOS, from the **Mobile Application Framework** tree node. This opens a page for the selected platform. Specify the platform and SDK location and then provide the information for the signing credentials.

For instructions on setting up development tools for the iOS and Android platforms, see:

- Installing Mobile Application Framework with JDeveloper, *Installing Oracle Mobile Application Framework*

- Setting Up the Development Environment, *Installing Oracle Mobile Application Framework*

Setting Up Oracle JDeveloper for Internationalization

In Oracle JDeveloper, select **Tools** from the menu, and then **Preference**.

In the Preference dialog, select **Environment** from the left panel, and then select "UTF-8" in the Encoding field from the drop-down list.

For information on internationalizing Oracle E-Business Suite mobile apps, see Internationalizing Oracle E-Business Suite Mobile Apps, page 8-1.

4

Using Mobile Application Archives for Enterprise Distribution

Introduction

This chapter explains the concept of enterprise distribution and provides the step-by-step instructions guiding you to use Mobile Application Archive (MAA) files to distribute mobile apps to internal users on an internal corporate location. It includes the following topics:

Understanding Enterprise Distribution

What is enterprise distribution?

It is the distribution of apps to mobile users through an enterprise-controlled location rather than through a public app store, such as Apple App Store or Google Play.

This capability allows enterprises to achieve the following main objectives, available from Oracle E-Business Suite Mobile Foundation Release 4.0:

- Version control

  Enterprises can control the version of the client apps that their enterprise users install on their mobile devices.

  See: Changing Application Bundle Id, page 4-16.

- Corporate branding

  Enterprises can have the option to replace the standard Oracle logos with their own company logos.

  See: Customizing Mobile Apps for Corporate Branding (Optional), page 4-23.

- Push notifications

  Starting from Oracle E-Business Suite Mobile Foundation Release 7.0 and onwards, enterprises can have the option to enable and send push notifications to the mobile devices of their enterprise users when using Oracle Mobile Cloud Service or using Oracle Mobile Hub from Release 9.0 and onwards. This feature is available for selected apps only. See Implementing Push Notifications, page 6-1.

  > **Note:** Starting from Oracle E-Business Suite Mobile Foundation Release 9.0 and onwards, Oracle Mobile Hub provides support for push notifications through Patch 33404902:R12.FND.C for Oracle E-Business Suite 12.2, and Patch 33404902:R12.FND.B for Oracle E-Business Suite 12.1.3.

Oracle E-Business Suite mobile apps allow you to perform certain customization; however, any additional changes beyond the content described in this book may not be supported.

With this capability, enterprise users can download the apps directly from an enterprise's internal location, such as an enterprise's own site, not from a public app store. Additionally, enterprises can only distribute these apps to their internal enterprise users through internal locations. These apps cannot be redistributed to a public app store or third-party users.

To accomplish these goals, Oracle E-Business Suite provides Mobile Application Archive (MAA) files for Oracle E-Business Suite mobile apps. Enterprises can use these MAA files for customization as allowed, generate their own application binaries such as iOS application bundle (.ipa) or Android application package (.apk), and deploy them to their own sites.

Understanding Mobile Application Archive Files

Oracle delivers mobile application archive (MAA) files for Oracle E-Business Suite mobile apps to enable changes for enterprise distribution.

An MAA file is an application archive that allows developers to use Oracle Mobile Application Framework (Oracle MAF) to customize the app and generate the enterprise version of the app to meet enterprise needs.

> **Important:** Oracle will provide technical support for issues that can be reproduced with MAA files delivered from Oracle and modified as

documented in this book. When Oracle makes changes or provides fixes for the mobile apps, the updates will be delivered as new MAA files, and you will need to reapply your changes to the latest files.

Oracle provides MAA files for most Oracle E-Business Suite mobile apps, including:

- Mobile apps built with Oracle E-Business Suite Mobile Foundation
  - Mobile apps available in Oracle E-Business Suite Mobile Release 9.1

    You can download the MAA files of the corresponding apps available in this release through My Oracle Support.

    See: Oracle E-Business Suite Mobile Application Archive for Release 9.1, page 4-6.

  - Mobile apps available in Oracle E-Business Suite Mobile Release 9.1 and earlier

    For mobile apps available in Oracle E-Business Suite Mobile Release 9.1 and earlier, you can download the associated MAA files through the Oracle Software Delivery Cloud. See: Mobile Application Archives for Earlier Oracle E-Business Suite Mobile Foundation Releases, page D-1.

    For downloading discontinued Oracle E-Business Suite mobile apps, see:

    - Discontinued Oracle E-Business Suite Mobile Apps in Oracle E-Business Suite Mobile Application Archive 9.1 Software Distribution, page 4-9
    - Discontinued Oracle E-Business Suite Mobile Apps in Oracle E-Business Suite Mobile Application Archive 7.0 Software Distribution, page 4-11

- Oracle Mobile Supply Chain Applications for Oracle E-Business Suite (MSCA)

  The associated MAA file for MSCA is available together with other mobile apps built with Oracle E-Business Suite Mobile Foundation 9.1. You can download the MAA file through Patch 32353571 available through My Oracle Support.

  See: Oracle E-Business Suite Mobile Application Archive for Release 9.1, page 4-6.

  Note that the information described in this chapter applies for mobile apps developed based on Oracle E-Business Suite Mobile Foundation. It also applies for the MSCA app with some exceptions where noted. For information on creating an enterprise version of the MSCA app, see Creating an Enterprise-Distributed App for Oracle Mobile Supply Chain Applications for Oracle E-Business Suite, page 4-33.

- Oracle Mobile Expenses for Oracle E-Business Suite (known as Oracle Fusion Expenses on the Apple App Store and Google Play)

  Oracle delivers MAA file for this app. For more information about this app, see My

Oracle Support Knowledge Document 1625446.1.

For information on creating enterprise-distributed mobile apps through the MAA files, see Creating Mobile Apps through MAA Files for Enterprise Distribution, page 4-4.

Creating Mobile Apps Through MAA Files for Enterprise Distribution

After setting up the development environment as described in the previous chapter, you can perform the following tasks to create enterprise-distributed apps from the mobile application archives:

1. Creating an Oracle JDeveloper Application from an MAA File, page 4-5
    1. Downloading Mobile Application Archives Files, page 4-5
    2. Importing an MAA File to Create a MAF Application, page 4-13
2. Updating the MAF Application with Required Changes, page 4-15
    1. Changing Application Bundle Id, page 4-16
    2. Changing the Privacy Policy Link, page 4-19
    3. Changing Mobile App Access Roles, page 4-21
3. Customizing Mobile Apps for Corporate Branding (Optional), page 4-23
4. Modifying an Existing Deployment Profile (Conditional), page 4-23
5. Updating Other Optional Application Configurations, page 4-29
    1. Updating Plug-in Configuration (Optional), page 4-29
    2. Enabling the Push Plug-in (Optional), page 4-29
    3. Importing Additional Root-CA Certificates (Optional), page 4-30
    4. Configuring Default Server URL (Optional), page 4-31
    5. Configuring Login Credentials in the Sign In Screen (Optional), page 4-32
6. Upgrading Your Enterprise Mobile Apps, page 4-32
7. Deploying Your Enterprise Mobile Apps, page 4-33

For information on creating an enterprise-distributed app for Oracle Mobile Supply Chain Applications for Oracle E-Business Suite, see Creating an Enterprise-Distributed App for Oracle Mobile Supply Chain Applications for Oracle E-Business Suite (MSCA),

page 4-33.

Creating an Oracle JDeveloper Application from an MAA File

Once a development environment is set up on the client side, you can download the MAA files and create an application from a downloaded MAA file.

To better understand each task performed on the client side, this section includes the following topics:

1. Downloading Mobile Application Archives Files, page 4-5
2. Importing an MAA File to Create a MAF Application, page 4-13

Downloading Mobile Application Archives Files

Use the following steps to download Oracle E-Business Suite Mobile Application Archive (.maa) files, along with consolidated server-side patches, Oracle E-Business Suite Mobile Foundation, and Oracle Mobile Application Framework 2.6.3 for Oracle EBusiness Suite Mobile Foundation Release 9.1.

> **Important:** Oracle is discontinuing selected Oracle E-Business Suite mobile apps. To download the corresponding MAA files for these discontinued apps for enterprise distribution and corporate branding, see: Discontinued Oracle E-Business Suite Mobile Apps in Oracle E-Business Suite Mobile Application Archive 9.1 Software Distribution, page 4-9 and Discontinued Oracle E-Business Suite Mobile Apps in Oracle E-Business Suite Mobile Application Archive 7.0 Software Distribution, page 4-11.

1. Log in to the Oracle Software Delivery Cloud (`https://edelivery.oracle.com/`) page.
2. Click the **Sign In** button.

   You are now signed in to the Oracle Software Delivery Cloud.

   If you do not have an Oracle account, click the **New User? Register Here** link instead to create one.
3. On the Oracle Service Delivery Cloud page, perform the following tasks:
   - Select "Release" from the drop-down list.
   - In the text field, enter "Oracle E-Business Suite Mobile Application Archive".

     The search results display the software distributions that meet your criteria. Click the **Select** button to choose your desired software distribution from the

populated package list. This action adds the selected package to the cart.

For example, select "Oracle E-Business Suite Mobile Application Archive 9.1".

- Click the **View Items** link.

  This displays the selected "Oracle E-Business Suite Mobile Application Archive 9.1" software distribution.

  Click the **Continue** button.

- Platform/Language: Select a desired platform, such as Linux x86-64 bit.

4. Click the **Continue** button and accept the Oracle Standard Terms and Restrictions.

   You must do so before you can download the desired media pack.

5. Click the **Continue** button again.

   This retrieves the download window with your selected software distribution.

   - To download the Mobile Application Archive (.maa) files for the apps developed based on Oracle E-Business Suite Mobile Foundation 9.1, including both current and discontinued apps, select and download the "Oracle E-Business Suite Mobile Application Archive 9.1" software distribution with part number `V1031711-01`.

     See: Oracle E-Business Suite Mobile Application Archive Files for Release 9.1, page 4-6.

   - To download the Mobile Application Archive (.maa) files for the discontinued apps developed based on Oracle E-Business Suite Mobile Foundation 7.0, select and download the "Oracle E-Business Suite Mobile Application Archive 7.0" software distribution with part number `V861706-01`.

     See: Discontinued Oracle E-Business Suite Mobile Apps in Oracle E-Business Suite Mobile Application Archive 7.0 Software Distribution, page 4-11.

For information on downloading mobile application archives files for earlier Oracle E-Business Suite Mobile Foundation releases, see Mobile Application Archives for Earlier Oracle E-Business Suite Mobile Foundation Releases, page D-1.

Oracle E-Business Suite Mobile Application Archive Files for Release 9.1

This "Oracle E-Business Suite Mobile Application Archive 9.1" software distribution (`V1031711-01`) is delivered in zip files. You should download the zip files to obtain the needed components for enterprise distribution and custom app development.

> **Note:** This software distribution also contains the Mobile Application Archive file for Oracle Mobile Supply Chain Applications for Oracle

EBusiness Suite (MSCA), although this app is not developed based on Oracle E-Business Suite Mobile Foundation.

This software distribution contains the following components:

- Consolidated server-side prerequisite patches

  For information about these server-side patches for the apps developed based on Oracle E-Business Suite Mobile Foundation and Oracle Mobile Supply Chain Applications for Oracle E-Business Suite (MSCA), see the Oracle E-Business Suite Mobile Foundation and Mobile Application Archive Release 9.1 Readme, included in this software distribution. Additionally, see Applying Prerequisite Patches on the Oracle E-Business Suite Server, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

- Oracle Mobile Application Framework 2.6.3 for Oracle E-Business Suite Mobile Foundation Release 9.1

  To work with downloaded Mobile Application Archive files, you need to use this Oracle MAF version 2.6.3 that is included in this software distribution. You can also download this MAF version from My Oracle Support through Patch 31581902.

- Oracle E-Business Suite Mobile Foundation package, for enabling custom app development for Oracle E-Business Suite

  For information about the Oracle E-Business Suite Mobile Foundation package and how to use it to develop custom apps, see Using the Login Component to Develop Mobile Apps, page 5-1.

- Mobile Application Archive (.maa) files delivered in zip files, along with the app-specific readme for each Oracle E-Business Suite mobile app included in this software distribution

  You can use the app-specific MAA file to customize the app for enterprise distribution and corporate branding, and then distribute the updated version of the app to your users through your enterprise's own site rather than a public app store.

  The following table lists the MAA file information associated with each Oracle E-Business Suite mobile app:

  > **Important:** Oracle E-Business Suite mobile apps may republish updated MAA files after a general availability release if necessary. Before downloading the MAA files listed in the following table from this software distribution, you should review the corresponding product release notes first to check for any updated MAA information. For information about product release notes, refer to My Oracle Support Knowledge Document 1641772.1, *Oracle E-Business Suite Mobile Apps, Release 12.1 and 12.2 Documentation*

Index.

Note: These MAA files built with Oracle Mobile Application Framework 2.6.3 for Oracle E-Business Suite Mobile Foundation Release 9.1 are also available for download through the associated patches from My Oracle Support.

Oracle E-Business Suite Mobile Application Archive Files for Oracle E-Business Suite Mobile Foundation 9.1

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile Approvals for Oracle E-Business Suite (**Footnote 1**, page 4-9)	Patch 32345543	p32345543_R12_GENERIC.zip
Oracle Mobile Discrete Production Supervisor for Oracle E-Business Suite (**Footnote 2**, page 4-9)	Patch 32358556	p32358556_R12_GENERIC.zip
Oracle Mobile Inventory for Oracle E-Business Suite (**Footnote 2**, page 4-9)	Patch 32348102	p32348102_R12_GENERIC.zip
Oracle Mobile Learning for Oracle E-Business Suite (**Footnote 2**, page 4-9)	Patch 32336727	p32336727_R12_GENERIC.zip
Oracle Mobile Maintenance for Oracle E-Business Suite	Patch 32365883	p32365883_R12_GENERIC.zip
Oracle Mobile Person Directory for Oracle E-Business Suite (**Footnote 2**, page 4-9)	Patch 32338590	p32338590_R12_GENERIC.zip
Oracle Mobile Process Production Supervisor for Oracle E-Business Suite (**Footnote 2**, page 4-9)	Patch 32359457	p32359457_R12_GENERIC.zip

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile iProcurement for Oracle E-Business Suite (**Footnote 2**, page 4-9)	Patch 32391756	p32391756_R12_GENERIC.zip
Oracle Mobile Sales Orders for Oracle E-Business Suite (**Footnote 2**, page 4-9)	Patch 32353121	p32353121_R12_GENERIC.zip
Oracle Mobile Self-Service Human Resources for Oracle E-Business Suite	Patch 32338602	p32338602_R12_GENERIC.zip
Oracle Mobile Timecards for Oracle E-Business Suite	Patch 32336857	p32336857_R12_GENERIC.zip
Oracle Mobile Yard for Oracle E-Business Suite (**Footnote 2**, page 4-9)	Patch 32353578	p32353578_R12_GENERIC.zip
Oracle Mobile Supply Chain for Oracle E-Business Suite (**Footnote 3**, page 4-9)	Patch 32353571	p32353571_R12_GENERIC.zip

Footnote 1: If you plan to distribute the Approvals app through enterprise distribution from the associated MAA file, you can optionally enable push notifications when using Oracle Mobile Hub or Oracle Mobile Cloud Service. For more details about implementing this feature for the mobile app, see Implementing Push Notifications, page 6-1.

Footnote 2: Oracle is discontinuing selected Oracle E-Business Suite mobile apps built with Oracle E-Business Suite Mobile Foundation 9.1. See: Discontinued Oracle E-Business Suite Mobile Apps in Oracle E-Business Suite Mobile Application Archive 9.1 Software Distribution, page 4-9.

Footnote 3: This software distribution also contains the MAA file for Oracle Mobile Supply Chain for Oracle E-Business Suite, although this app is not built with Oracle E-Business Suite Mobile Foundation.

Discontinued Oracle E-Business Suite Mobile Apps in Oracle E-Business Suite Mobile Application Archive 9.1 Software Distribution

Oracle is discontinuing selected Oracle E-Business Suite mobile apps as of November 1, 2022. Oracle will support these apps, but will not deliver any new updates to the apps.

- The apps will remain on the Apple App Store and Google Play with the latest client version delivered, as long as technically feasible. Apple and Google may remove apps that no longer meet their technical requirements.
- The corresponding MAA files for the latest client version are available on the Oracle Software Delivery Cloud.
- Oracle will continue to deliver REST services for selected functionality in these apps for use in custom app development.

Customers who have installed these apps will continue to receive technical support including access to online support tools, knowledge bases, pre-existing fixes, and service request resolution. As of November 1, 2022, error correction support will no longer be available for these apps.

Additionally, Oracle will not certify these apps with any further operating system updates, new devices, or new Oracle E-Business Suite releases after its discontinuation date. Oracle will continue to assist you to the best of our ability; however, we are unable to provide any new fixes.

The following table lists the MAA files for the discontinued Oracle E-Business Suite mobile apps as of November 1, 2022.

Discontinued Oracle E-Business Suite Mobile Apps in Oracle E-Business Suite Mobile Application Archive 9.1 Software Distribution

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile Discrete Production Supervisor for Oracle E-Business Suite	Patch 32358556	p32358556_R12_GENERIC.zip
Oracle Mobile Inventory for Oracle E-Business Suite	Patch 32348102	p32348102_R12_GENERIC.zip
Oracle Mobile Learning for Oracle E-Business Suite	Patch 32336727	p32336727_R12_GENERIC.zip
Oracle Mobile Person Directory for Oracle E-Business Suite	Patch 32338590	p32338590_R12_GENERIC.zip
Oracle Mobile Process Production Supervisor for Oracle E-Business Suite	Patch 32359457	p32359457_R12_GENERIC.zip
Oracle Mobile iProcurement for Oracle E-Business Suite	Patch 32391756	p32391756_R12_GENERIC.zip

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile Sales Orders for Oracle E-Business Suite	Patch 32353121	p32353121_R12_GENERIC.zip
Oracle Mobile Yard for Oracle E-Business Suite	Patch 32353578	p32353578_R12_GENERIC.zip

For instructions on downloading these discontinued apps contained in Oracle E-Business Suite Mobile Application Archive 9.1 Software Distribution, see Oracle E-Business Suite Mobile Application Archive Files for Release 9.1, page 4-6.

Discontinued Oracle E-Business Suite Mobile Apps in Oracle E-Business Suite Mobile Application Archive 7.0 Software Distribution

Oracle is discontinuing selected Oracle E-Business Suite mobile apps as of August 3, 2018. Oracle will support these apps, but will not deliver any new updates to the apps.

- The apps will remain on the Apple App Store and Google Play with the latest client version delivered, as long as technically feasible. Apple and Google may remove apps that no longer meet their technical requirements.
- Customers who have installed this app will continue to receive technical support including access to online support tools, knowledge bases, pre-existing fixes, and service request resolution. As of August 3, 2018, error correction support will no longer be available for this app.
- Additionally, Oracle will not certify this app with any further operating system updates, new devices, or new Oracle E-Business Suite releases after its discontinuation date. Oracle will continue to assist you to the best of our ability; however, we are unable to provide any new fixes.

For more information about these discontinued apps, see the Discontinued Oracle E-Business Mobile Apps section in *Oracle E-Business Suite Mobile Apps, Release 12.1 and 12.2 Documentation Index*, My Oracle Support Knowledge Document 1641772.1.

Downloading the "Oracle E-Business Suite Mobile Application Archive 7.0" Software Distribution

Follow the instructions described earlier to download the software distribution from the Oracle Software Delivery Cloud. In step 3, select "Oracle E-Business Suite Mobile Application Archive 7.0" (`V861706-01`) software distribution instead.

Similar to the "Oracle E-Business Suite Mobile Application Archive 9.1" software distribution described earlier, this software distribution contains the following components:

- Consolidated server-side prerequisite patches
- Oracle Mobile Application Framework 2.4.0 for Oracle E-Business Suite Mobile Foundation Release 7.0
- Oracle E-Business Suite Mobile Foundation package, for enabling custom app development for Oracle E-Business Suite
- Mobile Application Archive (.maa) files delivered in zip files, along with the app-specific readme for each Oracle E-Business Suite mobile app included in this software distribution

The following table lists ***only*** the associated MAA files for the discontinued apps described in this section.

Discontinued Oracle E-Business Suite Mobile Application Archive Files, built with Oracle E-Business Suite Mobile Foundation 7.0

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile Discrete Quality Manager for Oracle E-Business Suite	Patch 26160018	p26160018_R12_GENERIC.zip
Oracle Mobile Process Quality Manager for Oracle E-Business Suite	Patch 26180810	p26180810_R12_GENERIC.zip
Oracle Mobile Procurement for Oracle E-Business Suite	Patch 26197483	p26197483_R12_GENERIC.zip
Oracle Mobile Product Information for Oracle E-Business Suite	Patch 25949274	p25949274_R12_GENERIC.zip
Oracle Mobile Project Manufacturing for Oracle E-Business Suite	Patch 25933918	p25933918_R12_GENERIC.zip
Oracle Mobile Project Manager for Oracle E-Business Suite	Patch 25949676	p25949676_R12_GENERIC.zip

For more information about this "Oracle E-Business Suite Mobile Application Archive 7.0" software distribution and the included components described above, see Mobile Application Archives for Oracle E-Business Suite Mobile Foundation Release 7.0, page

D-10.

Importing an MAA File to Create a MAF Application

Use the following steps to create a MAF application by importing a downloaded MAA file:

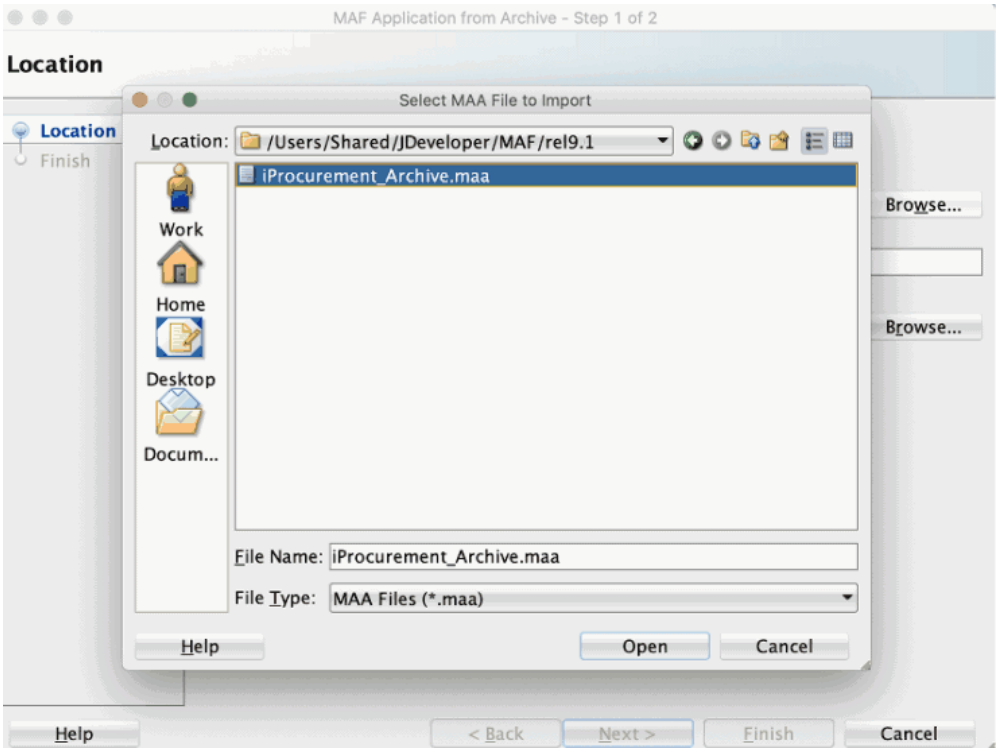
1. In Oracle JDeveloper, choose **File** and then **New**.

2. In the New Gallery, choose **Applications** and then **MAF Application from Archive File** and click **OK**.

   Alternatively, choose **File**, then **File Import**, and then select **MAF Application from Archive File**.

3. In the Location page, choose **Browse** in the MAA File field to locate the .maa file (such as "iProcurement_Archive.maa") to be imported in the Select MAA File to Import page. Click **Open**.

Select MAA File to Import Page

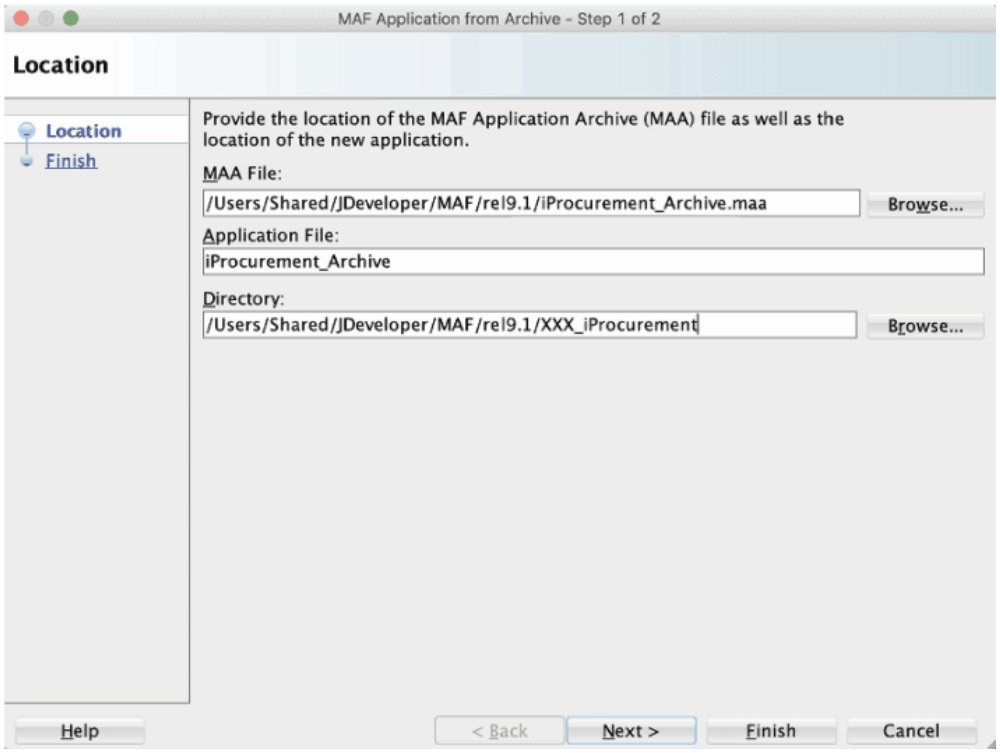


Note: The screenshots in this guide show the appearance of the pages on a Mac system. The look and feel may vary on a PC.

The selected iProcurement_Archive.maa file is displayed in the MAA File field.

4. Perform the following steps if needed or accept the default values in the Location page:

MAF Application from Archive: Location Page



1. In the Application File field, enter a name of the mobile application derived from .maa file, such as "XXX_iProcurement".

2. In the Directory field, click **Browse** to retrieve the directory of the mobile application.

5. Click **Next**.

6. Review the import summary information and then click **Finish**.

   A new MAF application is created.

For information on creating an unsigned application and what happens after importing an MAA file, refer to Creating Unsigned Deployment Packages, *Developing Mobile Applications with Oracle Mobile Application Framework*.

Updating the MAF Application with Required Changes

For enterprise distribution, even if no change is planned to an MAA file but simply distribute the associated app to an enterprise's own site, it is still required to make certain changes to the app. For example, the Application Bundle Id must be unique for each app installed on an iOS or Android device, although the app content itself is

exactly the same.

Specifically, you need to update the MAF application with the following required changes:

1. Changing Application Bundle Id, page 4-16
2. Changing the Privacy Policy Link, page 4-19
3. Changing Mobile App Access Roles, page 4-21

Changing Application Bundle Id

Changing Application Bundle Id is internal to an app, but it helps technically differentiate an enterprise app from a public one in the backend.

Instructions to Change the Application Bundle Id

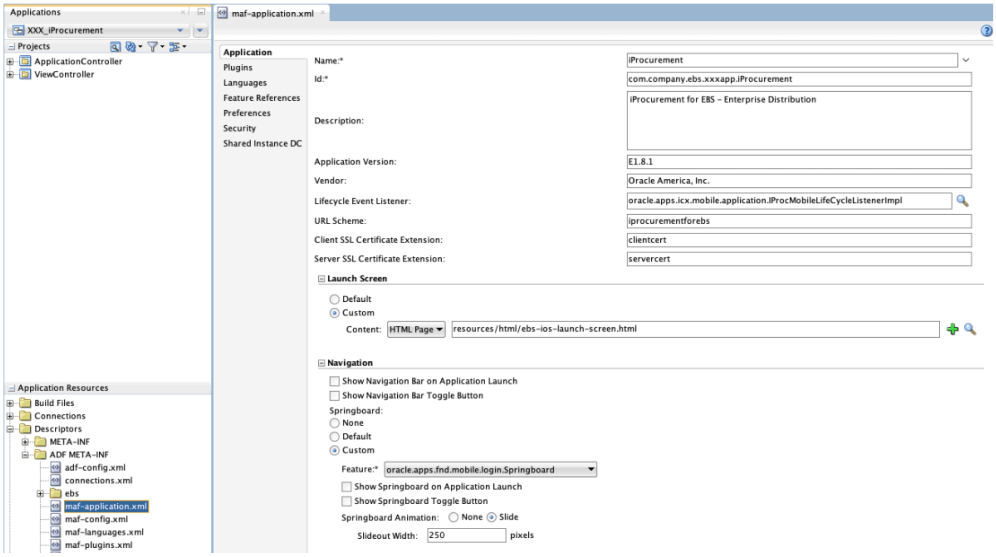
Perform the following steps to change the Application Bundle Id:

1. Open Oracle JDeveloper.
2. In the Applications Navigator, expand the **Application Resources** panel, then the **Descriptors** folder, and then the **ADF META-INF** folder.

   Double-click the **`maf-application.xml`** file to open the overview editor for the **`maf-application.xml`** file.

   For information on the **`maf-application.xml`** file, see About the MAF Application Feature Configuration File, MAF Application and Project Files, *Developing Mobile Applications with Oracle Mobile Application Framework*.

Overview Editor for the maf-application.xml File



3. Modify the Id field in the **`maf-application.xml`** file. Do not change any other fields in this file.

   Important: For enterprise distribution, the Application Version delivered in the MAA file for each app is prefixed with a letter "E" to indicate this app is created from the MAA file. For example, "E1.8.1" is the Application Version shipped in the MAA file for a seeded app version 1.8.1, built with Oracle E-Business Suite Mobile Foundation Release 9.1. This helps distinguish the enterprise-distributed apps from the publicly-distributed ones. This app version information is displayed in the About page for each app.

   To keep track of the enterprise app versions used in your company, you must continue to use the version provided in the MAA file, but you can add additional decimals at the end of the version.

   - Standard MAA version: E1.8.1

     This is the version of the corresponding out-of-the-box apps from pubic stores, but prefixed with a letter "E". In this example, 1.8.1 represents the version of the seeded apps built with Oracle E-Business Suite Mobile Foundation Release 9.1.

     For Oracle Mobile Supply Chain Applications for Oracle E-Business Suite, standard MAA version is E1.4.4.

   - Suggested customized version: E1.8.1.x.x, such as E1.8.1.1.0

For Oracle Mobile Supply Chain Applications for Oracle E-Business Suite, suggested customized version is E1.4.4.x.x, such as E1.4.4.1.0.

Id: Replace the Id with a unique Id to identify the mobile app for enterprise distribution.

The downloaded Oracle E-Business Suite mobile app MAA file contains the Application Bundle Id in the following format: `com.company.ebs.<prodfamily>.<product>.<AppName>`

For example, the Bundle Id from the MAA file corresponding to Oracle Mobile iProcurement for Oracle E-Business Suite is `com.company.ebs.prc.icx.iProcurement`.

In this example, use `com.company.ebs.xxxapp.iProcurement` as the Id.

- `company`- This can be replaced with your company name.
- `xxxapp`- It indicates to which Oracle E-Business Suite application it belongs.

Important: For mobile apps developed based on Oracle E-Business Suite Mobile Foundation, the Id value in the **`maf-application.xml`** file is used to download configuration details for the app. Therefore, use this Id as the Application Bundle Id value in the Application Details page when registering an enterprise app on the Oracle E-Business Suite server through the Mobile Applications Manager responsibility. See: Registering and Updating Your Mobile App Definition Metadata, page 3-5. If the registration corresponding to this Id is not found on the server, the associated mobile app cannot connect to that Oracle E-Business Suite instance.

This note does not apply for Oracle Mobile Supply Chain Applications for Oracle E-Business Suite.

Administrators use the Application Bundle Id to construct and validate the configuration service URL to ensure the app can be accessible from mobile users. For information on how to construct the configuration service URL using the Application Bundle Id, see Validating the Configuration, Setting Up the Mobile Apps, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Additionally, use the Application Bundle Id to help diagnose and troubleshoot any potential issues if occur on a mobile client. See: Enabling Client Logging, Diagnostics and Troubleshooting, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

4. Leave the rest of the fields unchanged. Save your work.

Changing the Privacy Policy Link

The privacy policy change is reflected in the About page of the app. After the change, when a user clicks the link, it should point to your company's privacy policy, not Oracle's Privacy Policy.

Instructions to Change the Privacy Policy Link

Perform the following steps to change the privacy policy URL link:

1. Open Oracle JDeveloper.

2. In the Applications Navigator, expand the **Application Resources** panel, then the **Descriptors** folder, then the **ADF META-INF** folder, and then the **ebs** folder.

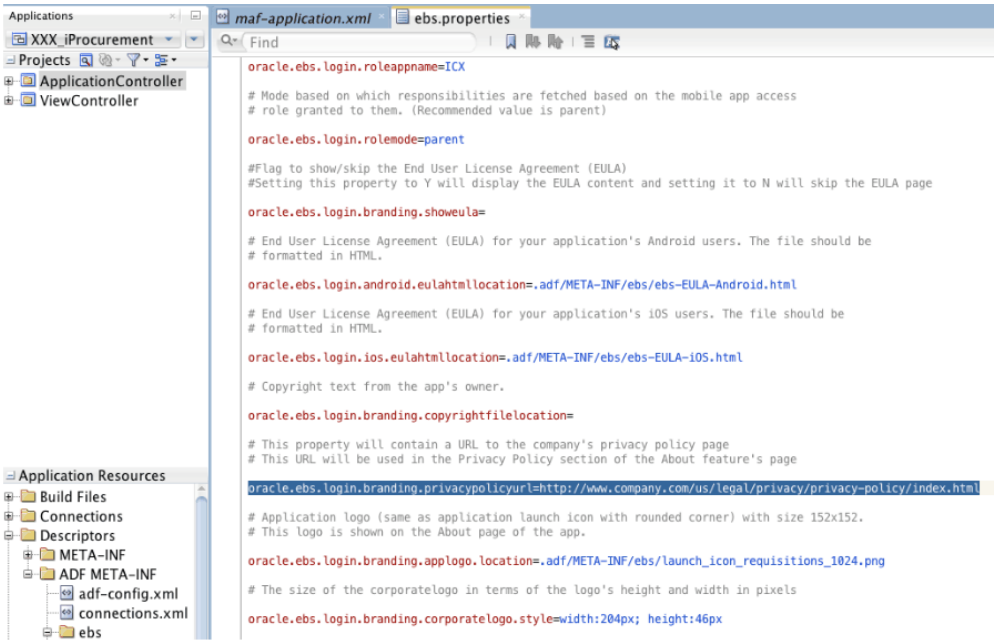
   Double-click the **`ebs.properties`** file to open it in an editor.

3. Replace the following privacy policy URL in the **`ebs.properties`** file with your company's policy URL, such as

   `oracle.ebs.login.branding.privacypolicyurl``=http://www.example.com/privacy-policy.html`.

   Please note that the downloaded Oracle E-Business Suite mobile app MAA file contains the following dummy URL: `http://www.company.com/privacy-policy.html`

Privacy Policy URL Property in ebs.properties

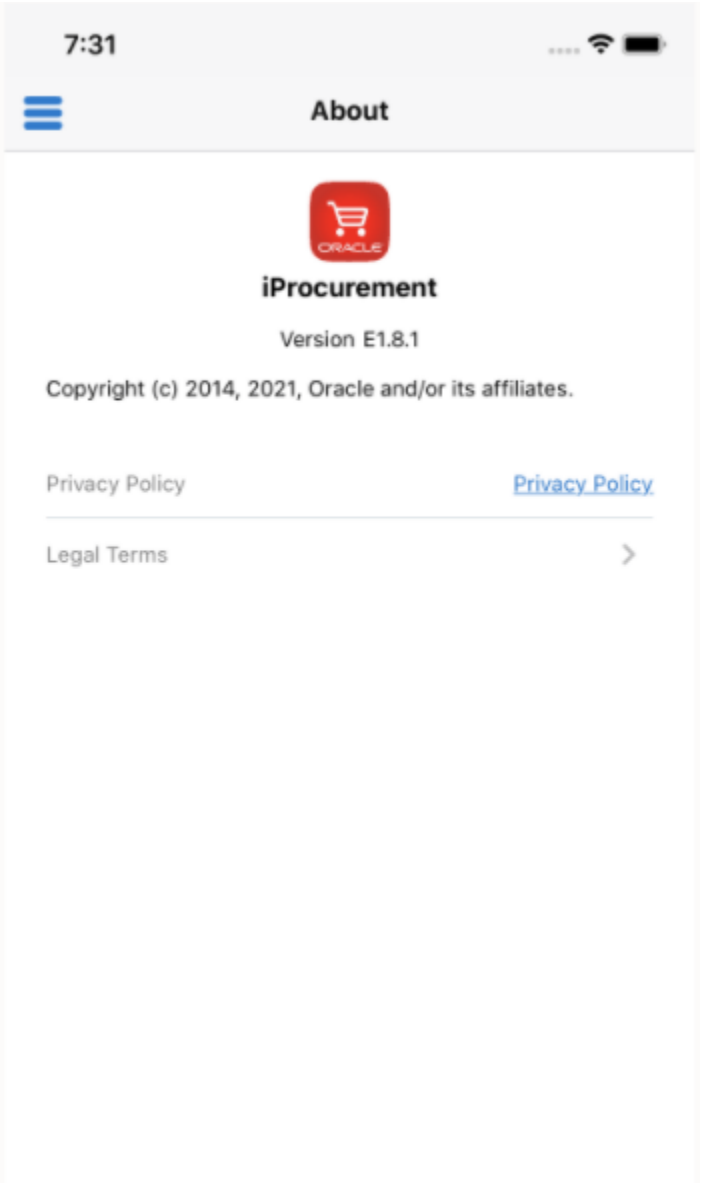


4. Save your work.

The following example shows the revised privacy policy URL link in the About page for the iProcurement app (Oracle Mobile iProcurement for Oracle E-Business Suite) on an iOS device.

In this example, nothing is changed in the app from the associated MAA file except that the privacy policy URL link is changed from pointing to a dummy URL to your company's privacy policy URL.

Privacy Policy Link in the About Page



Changing Mobile App Access Roles

Oracle E-Business Suite mobile apps use role-based access control to allow users who are assigned the appropriate access roles to access Oracle E-Business Suite.

Access roles are set up in the MAF application's `ebs.properties` file for each app to validate whether a mobile user has the privileges to access a designated app and connect to Oracle E-Business Suite. Therefore, once mobile app access roles are created in Oracle E-Business Suite, developers need to specify the corresponding access roles in the `ebs.properties` file for the mobile apps.

For information on creating the mobile app access roles on the Oracle E-Business Suite server, see Creating Mobile App Access Roles, page 3-19.

Perform the following steps to add the mobile app access roles in the `ebs.properties` file:

1. Open Oracle JDeveloper.
2. In the Application Navigator, expand the **Application Resources** panel.
3. Expand the **Descriptors** folder node and then the **ADF META-INF** folder node. Expand the **ebs** folder node.

   Double-click the **`ebs.properties`** file to open it in an editor.
4. Change the following properties to specify the role that your enterprise app (such as XXX_iProcurement) uses:
   - **`oracle.ebs.login.rolecode=`**`<UMX Role Code>`

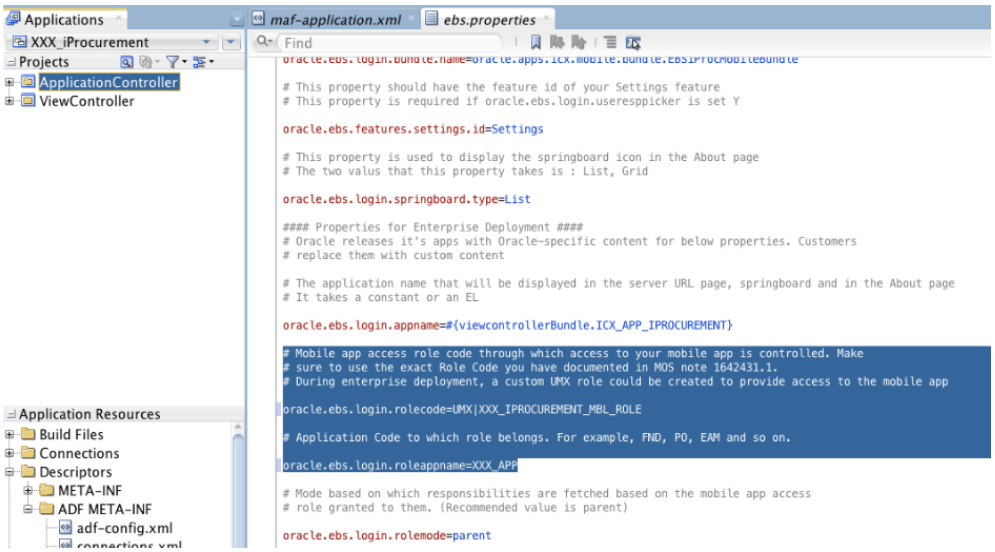
     If `UMX|XXX_IPROCUREMENT_MBL_ROLE` is a new role created earlier under the custom application called `XXX`, then replace `<UMX Role Code>` with this role `UMX|XXX_IPROCUREMENT_MBL_ROLE` in the property as shown here:

     `oracle.ebs.login.rolecode=``UMX|XXX_IPROCUREMENT_MBL_ROLE`
   - **`oracle.ebs.login.roleappname=`**`<Application Code>`

     Replace `<Application Code>` with `XXX_APP` in the property as shown here:

     `oracle.ebs.login.roleappname=``XXX_APP`

Mobile App Access Role Properties in ebs.properties



5. Save your work.

Customizing Mobile Apps for Corporate Branding (Optional)

In addition to the required changes for enterprise app creation, you can further implement corporate branding for the app to establish corporate identity by replacing Oracle logo with your own company logo.

Instructions on replacing Oracle logo with your company and app logos or images are documented in a separate chapter in this book. See: Implementing Corporate Branding, page 9-1.

> **Note:** Please note that functional customization and personalization of the apps modified from MAA files are not supported in this release.

Modifying an Existing Deployment Profile (Conditional)

After modifying the required changes for enterprise distribution, you need to prepare the platform-specific deployment profile by editing an existing deployment profile that is associated with the app provided with the MAA file.

> **Important:** If you have customized the mobile app for corporate branding as described in the previous section, you must have already created a new deployment profile. In this situation, skip this step and

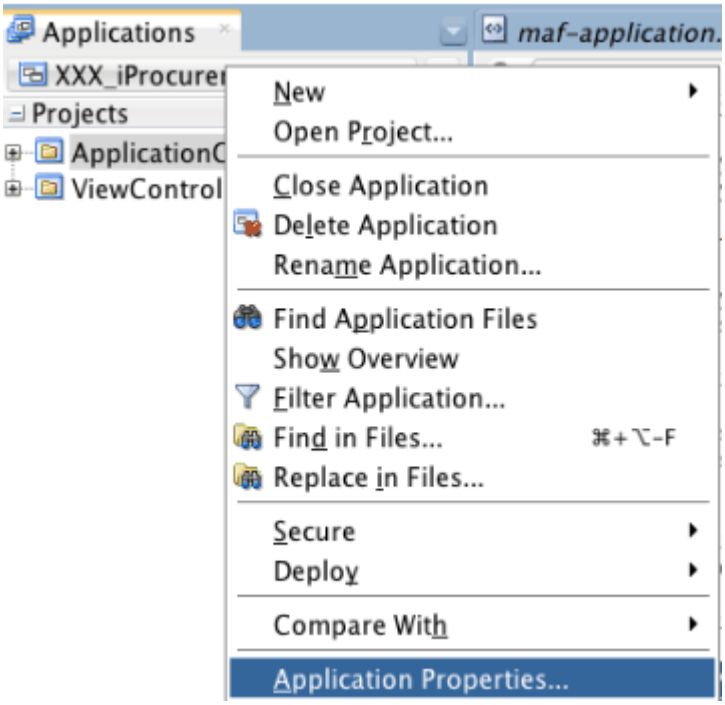
do not modify an existing deployment profile.

Please note that a deployment profile defines how an app will be deployed to iOS or Android powered devices, iOS simulators, or Android emulators.

Perform the following steps to modify an existing deployment profile for your desired platform:

1. In the Applications Navigator of Oracle JDeveloper, select and right-click the **Application**. Choose **Application Properties** from the selection window.

Oracle JDeveloper Navigation Menu with Application Properties Selected

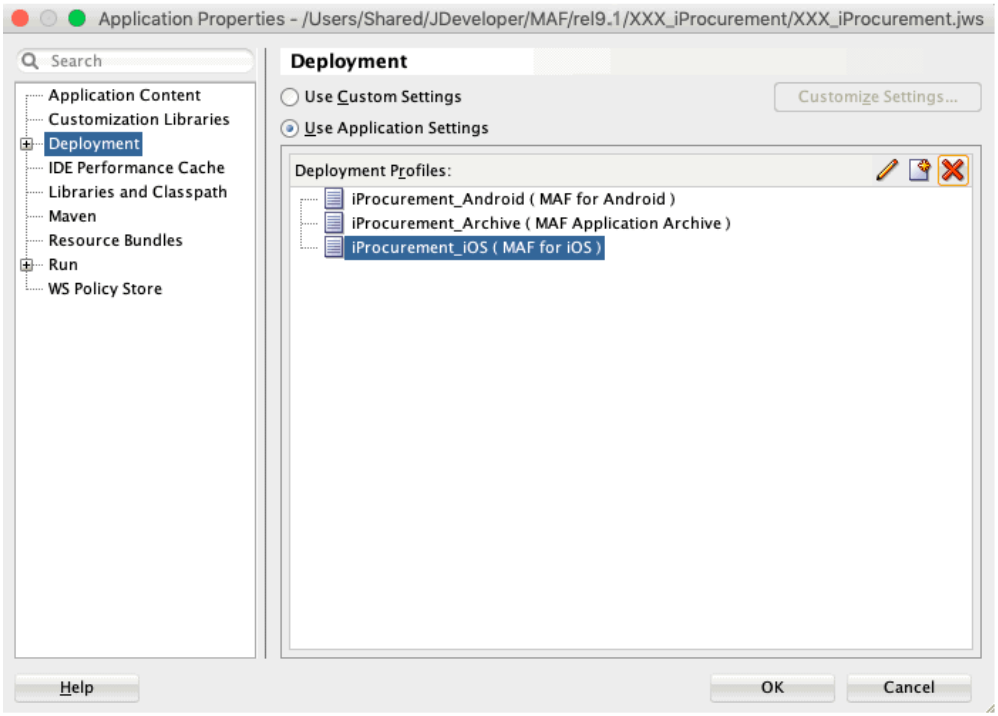


2. In the Application Properties window, select the **Deployment** node from the left pane.

   In the Deployment Profiles region, select the deployment profile you want to edit based on the platform and click on the **Edit** icon.

   > **Note:** For each mobile app, there are three deployment profiles delivered with the MAF archives, specifically one for the iOS platform (<AppName>_iOS), another for the Android platform (<AppName>_Android), and the other for the MAA file (<AppName>_Archive).

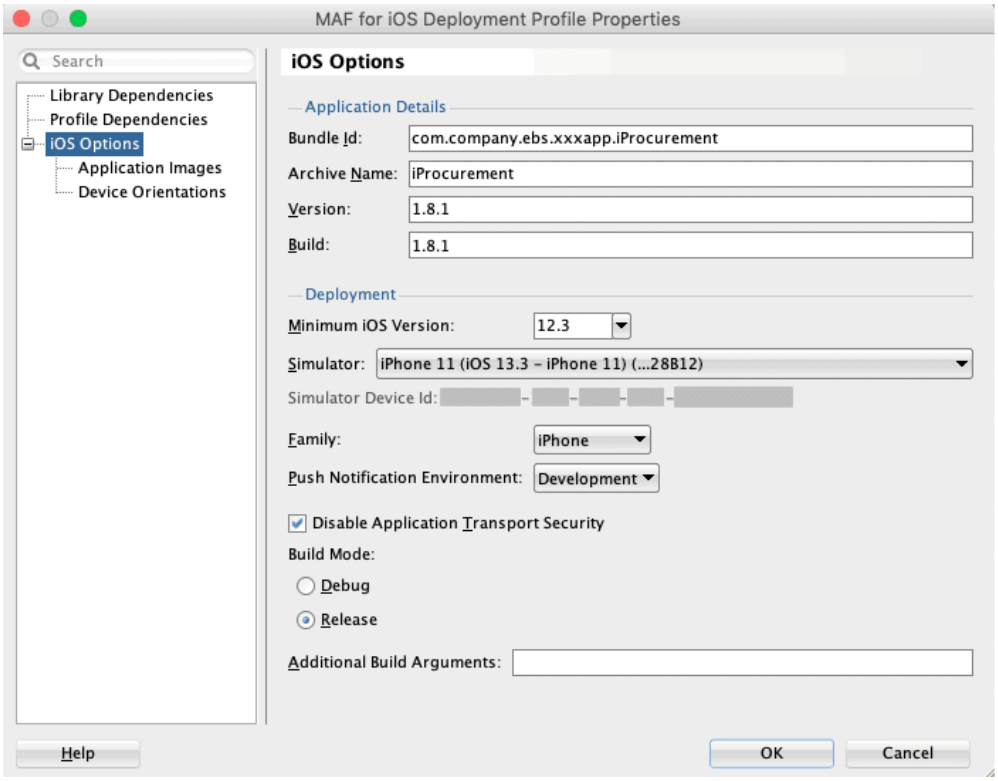
Application Properties Window with a Selected Deployment Profile



3. **Editing the MAF for iOS Deployment Profile**

   If the selected deployment profile (such as iProcurement_iOS) is for the iOS platform, the MAF for iOS Deployment Profile Properties page appears.

MAF for iOS Deployment Profile Properties Page



Replace the Application Bundle Id with a unique Id for your app, such as `com.company.ebs.xxxapp.iProcurement`.

- If you deploy the app to an iOS device, the Application Bundle Id should match the iOS provisioning profile that you receive after registering with Apple's iOS Developer Program.

  > **Note:** If the app supports push notifications, such as an enterprise-distributed Approvals app, the Application Bundle Id used here in the iOS deployment profile is also used when creating the iOS mobile client in Oracle Mobile Hub or Oracle Mobile Cloud Service. See: Creating Mobile Clients, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.
  >
  > For more information about push notifications, see Implementing Push Notifications, page 6-1.

- If you deploy the app to an iOS simulator for testing the mobile app, you can use any Application Bundle Id of your choice but you could use the same

Application Bundle Id required by the iOS provisioning profile.

The Application Bundle Id is used to package the application binary for iOS. Please note that the Application Bundle Id must be unique for each app installed on an iOS or Android device. Even if the same app has been deployed twice for an enterprise, a different Bundle Id uniquely represents each individual app. For example, one app can be used for production and the other one can be for testing purposes if desired.

Additionally, the Application Bundle Id used here impacts only the application binary packaging and its installation on mobile devices. It does not have any impact on the Oracle E-Business Suite server. For example, you could have packaged two apps with two different Application Bundle Ids in the MAF deployment profiles, such as `com.company.ebs.xxxapp.XXiProcurement` and `com.company.ebs.xxxapp.YYiProcurement`, but both could have the same Id in the **`maf-application.xml`** file, such as `com.company.ebs.xxxapp.iProcurement`. Both apps can be installed on the same mobile device and can connect to the same Oracle E-Business Suite server using single registration on the Oracle E-Business Suite server.

This is useful that you create the app registration once on the server, but you can download and apply it to other Oracle E-Business Suite instances. You can then use multiple installations of the same app on a single test device and test against different Oracle E-Business Suite instances.

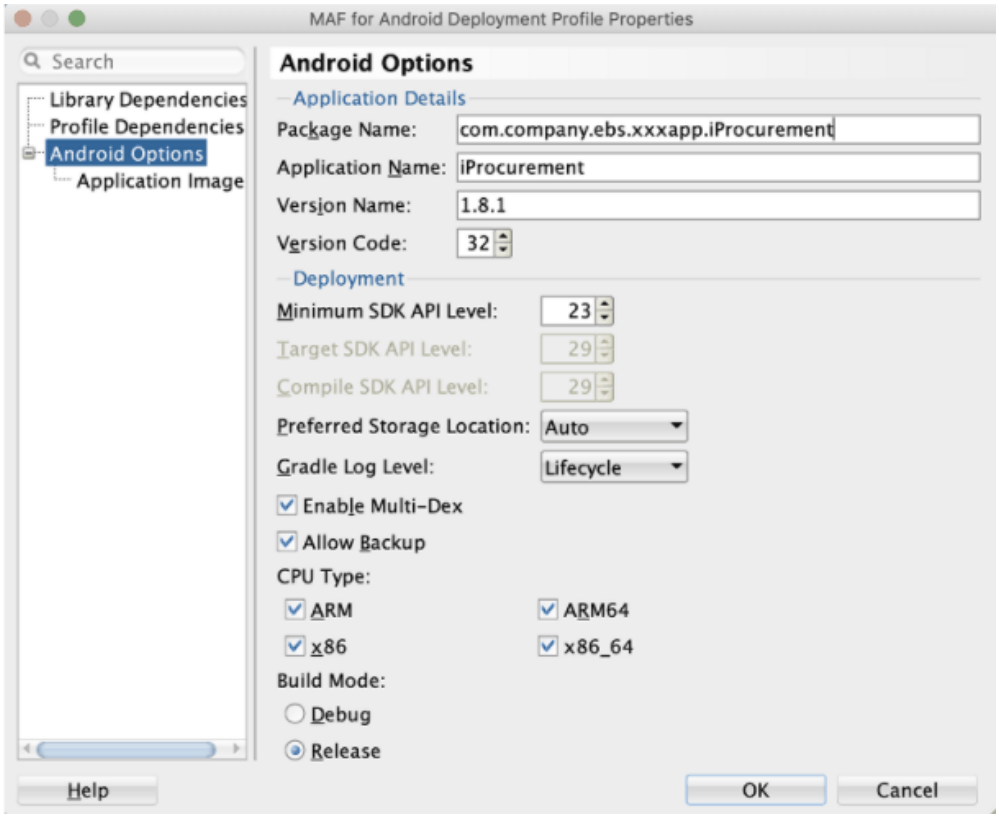
4. This Application Bundle Id is the only field you need to modify in this page. Click **OK**.

5. **Editing the MAF for Android Deployment Profile**

   If the selected deployment profile (such as iProcurement_Android) is for the Android platform, the MAF for Android Deployment Profile Properties page appears.

   Select "Application Details" from the Android Options tree node.

MAF for Android Deployment Profile Properties Page



Similar to the update for the iOS deployment profile, you only need to replace the Package Name with a unique Id, such as `com.company.ebs.xxxapp.iProcurement` in this page. Use the same value from the Id field that you modified in the **`maf-application.xml`** file, as described earlier in the Changing Application Bundle Id, page 4-16.

> **Note:** If the app supports push notifications, such as an enterprise-distributed Approvals app, the Package Name used here in the Android deployment profile is also used when creating the Android mobile client in Oracle Mobile Hub or Oracle Mobile Cloud Service. See: Creating Mobile Clients, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.
>
> For more information about push notifications, see Implementing Push Notifications, page 6-1.

Click **OK** to save your work.

For information on creating deployment profiles, see Working with Deployment Profiles, *Developing Mobile Applications with Oracle Mobile Application Framework*.

Updating Other Optional Application Configurations

You can optionally perform the following tasks if desired for your enterprise apps:

1. Updating Plugin Configuration (Optional), page 4-29
2. Enabling Push Plugin (Optional), page 4-29
3. Importing Additional Root-CA Certificates (Optional), page 4-30
4. Configuring Default Server URL (Optional), page 4-31
5. Configuring Login Credentials in the Sign In Screen (Optional), page 4-32

Updating Plug-in Configuration (Optional)

If your mobile apps include a Cordova plug-in, such as barcode scanner, to provide support for scanning barcodes to capture data, the related plug-in library is already packaged with the associated MAA files. For information about Oracle E-Business Suite mobile apps with barcode scanner, see Supporting for Barcodes, Setting Up the Mobile Apps, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

In order for your app to use the plug-in, before you deploy the app, perform the following steps to update the plug-in's path after creating the application:

1. Open Oracle JDeveloper.
2. In the Applications Navigator, expand the **Application Resources** panel, then the **Descriptors** folder, and then the **ADF META-INF** folder.
3. Double-click the **`maf-application.xml`** file.
4. In the overview editor that appears, click the Plugins navigation tab.
5. For the Cordova plug-in, update the path to point to the `<Application Root Folder>/src` folder where the barcode scanner is placed.

For more information on registering additional plug-ins in your MAF applications, see Using Plugins in MAF Applications, *Developing Mobile Applications with Oracle Mobile Application Framework*.

Enabling the Push Plug-in (Optional)

If your mobile app supports push notifications, you can optionally enable this feature by enabling the push plug-in to allow the app to receive push notifications on the mobile devices.

Note: Push notifications are currently supported only when using Oracle Mobile Hub or Oracle Mobile Cloud Service for the following apps. Note that in addition to Oracle Mobile Cloud Service, starting from Oracle E-Business Suite Mobile Foundation Release 9.0 and onwards, Oracle Mobile Hub provides support for push notifications when an appropriate patch is applied.

- Oracle Mobile Approvals for Oracle E-Business Suite, when provided to users through enterprise distribution
- Custom Oracle E-Business Suite mobile apps developed using the Login component from Oracle E-Business Suite Mobile Foundation

See: Enabling the Push Plug-in, page 6-7.

For more setup tasks to enable push notifications, see Implementing Push Notifications, page 6-1.

Importing Additional Root-CA Certificates (Optional)

For Oracle E-Business Suite mobile apps with enterprise distribution, custom certificates can be imported into cacerts of the mobile application. If your Oracle E-Business Suite environment is TLS-enabled, you can import additional root-CA certificates into your MAF application's truststore if the HTTPS server contains certificates not present in your MAF application's cacerts file.

Note: You do not need an Oracle MAF license to add custom certificates to a mobile application for enterprise distribution.

Note: Starting from Oracle E-Business Suite Mobile Foundation Release 7.0 and onwards, you can dynamically add custom CA or self-signed server certificates to the standard Oracle E-Business Suite apps, downloaded from the Apple App Store or Google Play, for TLS connections to an Oracle E-Business Suite server.

In the releases earlier than 7.0, to use custom CA or self-signed certificates, you may create a custom app with updated certificate list and distribute that app through enterprise distribution.

For information on updating and managing certificates in the cacerts file, see Migrating to New cacerts File for SSL in MAF 2.x.x, *Installing Oracle Mobile Application Framework*. For example, for mobile apps developed based on Oracle E-Business Suite Mobile Foundation 9.1, see the Oracle Mobile Application Framework 2.6.3 Migration Notes (https://www.oracle.com/application-development/technologies/maf/maf263migration.html) and "Migrating to New cacerts File for SSL in MAF 2.6.0".

For more information about support for custom CA or self-signed certificates and validating if the TLS certificate is valid or trusted, see Advanced Configurations for Secure Communication with HTTPS, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Configuring Default Server URL (Optional)

Starting from Oracle E-Business Suite Mobile Foundation Release 8.0 and onwards, you can preconfigure the Oracle E-Business Suite server URL that an app will use to connect to Oracle E-Business Suite. Once this configuration is complete, the app users no longer need to enter this URL manually after launching the app.

Depending on whether you use Enterprise Mobility Management (EMM) solutions, you can configure the default server URL in the following ways:

- If you do NOT use Enterprise Mobility Management (EMM) solutions, you can preconfigure the server URL in **`ebs.properties`** for the following apps only:
  - Apps provided to users through enterprise distribution
  - Custom apps developed based on Oracle E-Business Suite Mobile Foundation

  Configure the following properties in **`ebs.properties`** for these apps:

  - `oracle.ebs.login.server.url`

    This is the Oracle E-Business Suite server URL that an app should connect to by default. If a valid Oracle E-Business Suite server URL is entered in this property, the app users will not be prompted to enter a server URL when the app is launched for the first time.

    Ensure that your enterprise-distributed app or custom app built with Oracle E-Business Suite Mobile Foundation is already "Enabled" in the Mobile Applications Manager UI pages. For information on enabling an app, see Enabling a Mobile App Individually and Specifying the Configuration Through the UI Pages, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

  - `oracle.ebs.login.server.url.allow_change`

    If a default Oracle E-Business Suite server URL is used in the `oracle.ebs.login.server.url` property, you need to explicitly indicate whether the mobile users can change it in the app. By default, the users are not allowed to change the default URL. However, set it to "Y" only if you want to allow users to change the URL.

- If you plan to use Enterprise Mobility Management (EMM) solutions, administrators can preconfigure the server URL in an EMM console for the following apps:

- Standard apps installed from the Apple App Store or Google Play
- Apps provided to users through enterprise distribution
- Custom apps developed based on Oracle E-Business Suite Mobile Foundation

After the configuration, app users no longer need to enter the URL manually after launching an app installed from an EMM's app catalog. For the setup information in EMM, see Setup Tasks for Deploying Mobile Apps with Enterprise Mobility Management Solutions, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Note that once the configuration is performed in EMM, there is no need to set up the properties mentioned above in **`ebs.properties`**.

Configuring Login Credentials in the Sign In Screen (Optional)

Oracle E-Business Suite mobile apps have the "Remember User Name" feature turned on in the Sign In screen by default. When this feature is enabled, the user name is retrieved from cache and automatically displayed in the screen during the login process. The app user only needs to enter the password to log in to the app.

For enterprise distributed mobile apps, you can optionally update the login credentials configuration for the Sign In screen through the Create MAF Login Connection page in the **`maf-application.xml`** file. For example, you can disable this "Remember User Name" feature if you want the user name to be entered each time when a user logs in to an app or enable other automatically login features which have been turned off during the mobile app development if you want to enhance the user login experiences.

For information on configuring the login credentials in the Sign In screen, refer to "How to Store Login Credentials" from the Securing MAF Applications chapter in *Developing Mobile Applications with Oracle Mobile Application Framework*.

Upgrading Your Enterprise Mobile Apps

If you would like to upgrade your enterprise-distributed mobile app from a previous version of the corresponding mobile application archive (MAA) file to a new version, there is no direct upgrade path by retaining all the documented changes you completed previously. In order to upgrade the app to a new version, you should repeat the steps as described earlier in this chapter. These tasks include:

1. Creating an Oracle JDeveloper Application from an MAA File, page 4-5
2. Updating the MAF Application with Required Changes, page 4-15
3. Customizing Mobile Apps for Corporate Branding (Optional), page 4-23
4. Modifying an Existing Deployment Profile (Conditional), page 4-23

5. Updating Other Optional Application Configurations, page 4-29

As part of this upgrade to a new version of MAA, it is very important to use the exact same Application Bundle Id that was used in the previous version in **`maf-application.xml`** and the deployment profiles. Once you have completed the required changes to create an enterprise app from the new version of the MAA file, you can host it at the same location as the previous version and then coordinate with the mobile applications administrator for user upgrade.

Deploying Your Enterprise Mobile Apps

After completing the required changes for an enterprise app, you can deploy the app for the iOS, Android, or both platforms. For information on deploying your enterprise apps, see Deploying Your Mobile Apps, page 9-39.

Creating an Enterprise-Distributed App for Oracle Mobile Supply Chain Applications for Oracle E-Business Suite

Oracle Mobile Supply Chain for Oracle E-Business Suite (MSCA) mobile app supports enterprise distribution through the Mobile Application Archives (MAA) file.

Because this app is not built based on Oracle E-Business Suite Mobile Foundation, when you create an enterprise-distributed app from the MSCA mobile application archive, follow the instructions as described earlier in Creating Mobile Apps through MAA Files for Enterprise Distribution, page 4-4 ***except*** the following tasks that are not applicable to the enterprise version of the MSCA app:

> **Important:** Although the steps for creating an enterprise-distributed MSCA mobile app are not exactly the same as the tasks described earlier for creating enterprise distribution from the apps developed based on Oracle E-Business Suite Mobile Foundation, the concept of enterprise distribution discussed earlier remains the same for the MSCA app.

- Changing Mobile App Access Roles, page 4-21
- Enabling the Push Plugin (Optional), page 4-29
- Importing Additional Root-CA Certificates (Optional), page 4-30
- Configuring Login Credentials in the Sign In Screen (Optional), page 4-32

Note that you can implement corporate branding for the enterprise-distributed MSCA app. When replacing Oracle logo with your company logo by following the instructions described in Implementing Your Company Logo, page 9-27, this changes the company logo in the Server URL screen only. Additionally, the MSCA app does not leverage the

Oracle E-Business Suite Mobile Foundation Login component for custom app development.

For information on the underlying Oracle MAF version for your MSCA app, see Section 1: Release Update History, *Oracle Mobile Supply Chain Applications for Oracle E-Business Suite Release Notes*, My Oracle Support Knowledge Document 2108155.1.

5

Using the Login Component to Develop Mobile Apps

Overview

Oracle E-Business Suite Mobile Foundation Login Component (the Login component), available from Oracle E-Business Suite Mobile Foundation Release 4.1, is a library to help you easily develop and test mobile apps developed for Oracle E-Business Suite using Oracle Mobile Application Framework (MAF).

> **Note:** Starting from Oracle E-Business Suite Mobile Foundation Release 7.0, custom apps developed using the Login component can optionally be enabled with push notifications. Oracle E-Business Suite Mobile Foundation uses Oracle Mobile Hub, from Release 9.0 and onwards when an appropriate patch is applied, or Oracle Mobile Cloud Service to provide support for push notifications.
>
> For more details about implementing this feature for the mobile app, see Implementing Push Notifications, page 6-1.

The component implements the following features that are common to all Oracle E-Business Suite mobile apps developed based on Oracle E-Business Suite Mobile Foundation so that you can focus on implementing application functionality.

- Application Configuration
  - Enter server URL
  - Change server URL
  - Check for updates
- Sign In screen

- Security context
- Diagnostics
- About screen

To better understand how to create custom apps using the Login component or APIs, this chapter includes the following topics:

Understanding the Login Component Features

Oracle E-Business Suite mobile apps, like the self-service web applications, can be accessed only by authorized users of Oracle E-Business Suite. This requires a mobile app user to enter a user name and corresponding password, select an appropriate application responsibility as well as an organization if required before the app can successfully retrieve data from Oracle E-Business Suite. Unlike self-service web applications where the users are taken to a single Oracle E-Business Suite home page from where the users can access an application through its responsibility, mobile apps are designed and implemented for specific application functionalities. Hence, there is no Oracle E-Business Suite home page experience when implementing smartphone apps.

Before using the Login component to develop mobile apps against Oracle E-Business Suite, it is important to understand the following concepts:

Supported User Flows

The Login component provides the following user flows.

Initial Configuration and Context Initialization

The following diagram illustrates the process flow that a mobile app user goes through while attempting to use the app for the first time:

Mobile Apps Initial Launch Process Flow

During the initial launch of the app, the user is required to perform the following tasks:

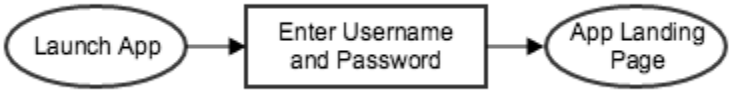
1. Accept End User License Agreement if the app requires one (optional).
2. Enter the Server URL to connect to an Oracle E-Business Suite instance.
3. Enter Oracle E-Business Suite user name and password in the Sign In screen.
4. Select a responsibility if the app requires one (optional).
5. Select an organization if the app requires one (optional).

For the tasks that are marked optional, the Login component allows you to remove the tasks in the flow using simple configuration. There are one-time activities in the diagram that are required to be done only when the app is launched for the first time.

Regular Login

Once the app is set up and used successfully for the first time, the user goes through the following flow for all subsequent uses of the app:

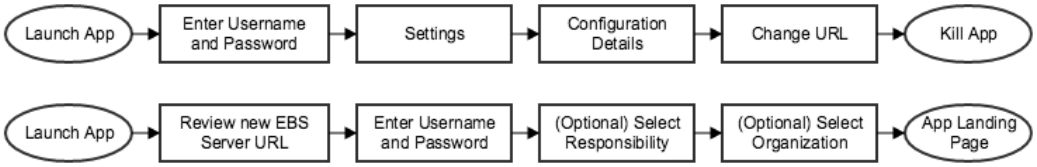
Mobile Apps Subsequent Launch Process Flow



Application Reconfiguration

There may be times when a user wants to use the app against a different Oracle E-Business Suite instance. This is the flow that the user goes through to change the Server URL to point to a different Oracle E-Business Suite instance.

Mobile Apps Launch Process Flow After Reconfiguration



The user is required to perform the following tasks:

1. Log in to the app. Go to the Settings screen and then the Connection Details screen.
2. Tap **Change URL** to replace the current server URL with a new server URL.
3. Force the app to close and then launch the app again.
4. Review the new server URL.

   In this step, you can decide to proceed with the new server URL or cancel the reconfiguration to use the previous server URL if desired.
5. Enter Oracle E-Business Suite user name and password in the Sign In screen.
6. Select a responsibility if the app requires one (optional).
7. Select an organization if the app requires one (optional).

Please note that steps that are marked optional are listed here only if the user originally went through them in the Initial Configuration and Context Initialization, page 5-3.

Mobile App Security

Oracle E-Business Suite mobile apps are secured against Oracle E-Business Suite security model for both Authentication and Authorization. Following are the security features implemented by the Login component to secure your mobile apps against Oracle E-Business Suite.

- Authentication
  - Authenticate users using existing Oracle E-Business Suite user credentials
  - Support the following authentication types:
    - Apps Local Login (previously known as "HTTP Basic") - for local authentication

      This authentication type corresponds to the "HTTP Basic" authentication server type used in Oracle Mobile Application Framework.

    - Apps SSO Login (previously known as "Web SSO") - for remote authentication

      This authentication type corresponds to the "Web SSO" authentication server type used in Oracle Mobile Application Framework.

- Authorization
  - Secure app access through mobile app access roles
  - Secure app data access through Oracle E-Business Suite's security context using responsibility and organization

Authentication

To access Oracle E-Business Suite mobile apps, mobile app users use the same user name and password information that they use to log in to Oracle E-Business Suite self-service applications. Administrators can still choose a specific authentication type for each app based on how the authentication is set up in Oracle E-Business Suite.

- Apps Local Login (previously known as "HTTP Basic") - for local authentication

  The "Apps Local Login" authentication is the default type for a mobile app to authenticate mobile users locally. When this type is selected for a mobile app, the users are authenticated locally against the Oracle E-Business Suite server.

- Apps SSO Login (previously known as "Web SSO") - for remote authentication

  When the "Apps SSO Login" type is selected for a mobile app, the mobile users are not authenticated against Oracle E-Business Suite, but against an external Oracle Access Manager (OAM) server.

  Use this authentication type if you want to delegate authentication to Oracle Access Manager based on a protected Login URL.

The authentication type for each mobile app is configured on the Oracle E-Business Suite server using the Mobile Applications Manager responsibility at the time the app is deployed for users. The authentication type value determines the configuration parameters required to set when configuring your mobile app.

Authorization

If an Oracle E-Business Suite mobile app requires responsibility context to access Oracle E-Business Suite data, then two levels of authorization setup are required:

- Mobile App Access Role - The mobile app itself is associated with a mobile app access role. The role is then granted to responsibilities that can access the mobile app.

- Responsibility Context - After signing in to the mobile app for the first time, a mobile user is given a list of responsibilities to which the app's access role is

granted. The user can select a responsibility and optionally also select an organization.

Configuration File

In order for the Login component to connect to an Oracle E-Business Suite instance and provide secure access to the data for the mobile apps, it requires a set of configuration parameters. The configuration parameters for a given mobile app are set up by the administrators on the Oracle E-Business Suite Server using the Mobile Applications Manager responsibility. The configuration parameters are then downloaded from the Oracle E-Business Suite server by the Login component based on the server URL provided by the user. The most important configuration parameter for a given mobile app is the Authentication Type. Based on the authentication type, the corresponding configuration parameters determine how the authentication works for the mobile app.

- Apps Local Login (previously known as "HTTP Basic") (local authentication)

  If "Apps Local Login" is selected as the authentication type to authenticate users locally against the Oracle E-Business Suite server, the associated configuration parameters are listed in the following table:

Apps Local Login Configuration Parameters

Parameter Name	Parameter Internal Name	Description	Default Value
Service Endpoint	APPS_MOBILE_AGENT	Oracle E-Business Suite server host URL to use for all web service invocations from a mobile app.	Value of the APPS_FRAMEWORK_AGENT profile option
Idle Timeout	APPS_MOBILE_IDLE_TIMEOUT	Time period after which the user is required to log in again to access the app feature. Unsaved data is not lost.	7200

Parameter Name	Parameter Internal Name	Description	Default Value
Session Timeout	APPS_MOBILE_SESSION_TIMEOUT	Time period after which the user's session is cleared and required to log in again to create a new session. Unsaved data is lost.	28800

- Apps SSO Login (previously known as "Web SSO") (remote authentication)

  If "Apps SSO Login" is selected as the authentication type to authenticate users remotely against an external Oracle Access Manager (OAM) server, the associated configuration parameters are listed in the following table:

Apps SSO Login Configuration Parameters

Parameter Name	Parameter Internal Name	Description	Default Value
SSO Login URL	LoginURL	This is the login server URL that challenges the user to authenticate with Oracle Access Manager (OAM). If the URL is valid, a mobile app displays the login screen where a user enters the credentials for user validation through Oracle Access Manager (OAM).	Value of the APPS_AUTH_AGENT profile option is suffixed with /login/sso

Parameter Name	Parameter Internal Name	Description	Default Value
SSO Logout URL	LogoutURL	This is the server-side URL that logs out a mobile user by terminating the server session from Oracle Access Manager.	Value of the APPS_AUTH_AGENT profile option is suffixed with /logout/sso
SSO Login Success URL	LoginSuccessURL	This is the URL that indicates the user has logged in successfully. To determine the correct value for this parameter, navigate to the configured SSO Login URL in a web browser session and then submit valid login credentials. The URL that you are re-directed to after successful login is your SSO Login Success URL. Please note that this URL can be the same as the SSO Login URL. In this release, the same URL is used for this SSO Login Success parameter and the SSO Login URL parameter, and it is the current value of Value of the APPS_AUTH_AGENT profile option suffixed with /logout/sso.	Value of the APPS_AUTH_AGENT profile option is suffixed with /logout/sso

Parameter Name	Parameter Internal Name	Description	Default Value
SSO Login Failure URL	LoginFailureURL	This is the URL to redirect a user to a login failure page after the authentication fails from the login page. This parameter is reserved for future use.	Value of the APPS_FRAMEWORK_AGENT profile option

Parameter Name	Parameter Internal Name	Description	Default Value
SSO Session Timeout	SessionTimeOutValue	The number of seconds that a user can remain logged in to an app. This parameter is specified in seconds, and the minimum value is 300 seconds. The default value is 28800 seconds. After the SSO session expires, the user will be prompted with the SSO login page. It is recommended that you set this parameter to a value that is less than the Oracle E-Business Suite session timeout value set in the ICX_SESSION_TIMEOUT profile option. This setting helps avoid issues with REST call failures after the ICX session timeout. For example, if the ICX_SESSION_TIMEOUT value is set to 30 minutes, you can set the SSO Session Timeout value to 1740 seconds (29 minutes). After the SSO session expires, the user will be prompted with the SSO login page.	28800

Parameter Name	Parameter Internal Name	Description	Default Value
EBS Session Service	APPS_SESSION_SERVICE	Oracle E-Business Suite service endpoint used to create the Oracle E-Business Suite server session based on OAM token.	Value of the APPS_AUTH_AGENT profile option is suffixed with /logout/apps
Service Endpoint	APPS_MOBILE_AGENT	Oracle E-Business Suite server host URL to use for all web service invocations from a mobile app.	Value of the APPS_FRAMEWORK_AGENT profile option

Developing Your Mobile Apps Using the Login Component

This section provides the step-by-step instructions on using the Login component to develop custom apps. Specifically, it includes the following topics:

- Step 10: Implementing Java Classes, page 5-44
- Step 11: Using Application Logging, page 5-44
- Step 12: Implementing Corporate Branding, page 5-45
- Step 13: Using Developer Mode, page 5-45

Starting from Oracle E-Business Suite Mobile Foundation Release 7.0 and onwards, you can optionally enable push notifications for your custom apps developed using the Login component. For information on implementing this feature for these custom apps, see Implementing Push Notifications, page 6-1.

Downloading and Using the Login Component

After applying the required patches for developing custom apps, as described in Applying Prerequisite Patches on the Oracle E-Business Suite Server, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*, and performing needed setup tasks on the server side, you can download and use the Login component for your custom app development.

The Oracle E-Business Suite Mobile Foundation Login component developed for Oracle E-Business Suite Mobile Foundation Release 9.1 is available for download from My Oracle Support through Patch 32284288. This patch applies to Oracle E-Business Suite Release 12.1.3 and Release 12.2.

> **Note:** For Oracle E-Business Suite Mobile Foundation (Login component) Release 9.0, you can download it through Patch 30914694. It is also included in the "Oracle E-Business Suite Mobile Application Archive 9.0" software distribution from the Oracle Software Delivery Cloud. For download instructions, see Downloading Mobile Application Archives Files, page 4-5. For Oracle E-Business Suite Mobile Foundation (Login component) Release 8.0, download it through Patch 27958894; for Oracle E-Business Suite Mobile Foundation (Login component) Release 7.0, download it through Patch 26023015.

You can use the downloaded Login component to develop custom apps for Oracle E-Business Suite. This component enables the Oracle E-Business Suite Mobile Foundation client libraries, application template, and sample app.

The following table lists the files contained in the Login component for you to use for custom app development:

Folder Name	Subfolder and File Name
maf/project This folder includes the Login component libraries and other Oracle Mobile Application Framework (Oracle MAF) project files. Files contained in this folder should be included into your Oracle MAF JDeveloper project.	lib • `EBSLoginCC.jar` • `EBSLoginLib.jar` adf/META-INF • `wsm-assembly.xml` • `maf-config.xml` • `connections.xml` adf/META-INF/ebs • `ebs.properties` • `Custom-EULA-Generic.html` • `CorporateLogo.png` src/META-INF • `logging.properties` • `maf.properties` src/Cordova-Plugin-InAppBrowser This folder contains Cordova plug-in implementation for InAppBrowser. ApplicationController/src/META-INF • `maf-skins.xml`

Folder Name	Subfolder and File Name

	ApplicationController/public_html/resources
	• `html/ebs-Login.html`
	• `images/CorporateLogo.png`
	• css
	  • `ebs-Login.css`
	  • `EBSMobile-1.1.css`
	  • `EBSMobile-1.1.Android.css`
	  • `EBSMobile-1.1.iOS.css`
	  • `EBSMobile-1.0.css`
	  • `EBSMobile-1.0.Android.css`
	  • `EBSMobile-1.0.iOS.css`
	• js
	  • `ebs-Login.js`
	  • `ebs-LoginBundle.js`
	  • `ebs-LoginBundle_de.js`
	  • `ebs-LoginBundle_es.js`
	  • `ebs-LoginBundle_es_US.js`
	  • `ebs-LoginBundle_fr.js`
	  • `ebs-LoginBundle_fr_CA.js`
	  • `ebs-LoginBundle_it.js`
	  • `ebs-LoginBundle_ja.js`
	  • `ebs-LoginBundle_nl.js`
	  • `ebs-LoginBundle_pt_BR.js`
	  • `ebs-LoginBundle_zh.js`

Folder Name	Subfolder and File Name

	ViewController/public_html/resources
	• js
	  • `ebs-ControlActions.js`
	• images
	  • `springboard_list.png`

maf/template This folder contains an Oracle JDeveloper project, built with Oracle Mobile Application Framework (MAF) 2.6.3 (or Oracle MAF 2.6.2 for Oracle E-Business Suite Mobile Foundation Release 9.0, Oracle MAF 2.5.0 for Oracle E-Business Suite Mobile Foundation Release 8.0, Oracle MAF 2.4.0 for Oracle E-Business Suite Mobile Foundation Release 7.0) and preconfigured with the Oracle E-Business Suite Login component. This project can be used as a template to quickly start developing custom mobile apps against Oracle E-Business Suite.	• `EBSMobileAppTemplate.zip`

Folder Name	Subfolder and File Name
maf/sample This folder contains an Oracle JDeveloper project, built with Oracle Mobile Application Framework 2.6.3 (or Oracle MAF 2.6.2 for Oracle E-Business Suite Mobile Foundation Release 9.0, Oracle MAF 2.5.0 for Oracle E-Business Suite Mobile Foundation Release 8.0, Oracle MAF 2.4.0 for Oracle E-Business Suite Mobile Foundation Release 7.0) which is a sample mobile app that is implemented using the Login component that can connect to Oracle E-Business Suite. Please note that the sample app acts as a reference implementation for: • Developing an Oracle E-Business Suite mobile app using the Login component • Implementing a List of Values (LOV) • Implementing a simple search box over List View • Implementing error, warning and confirmation messages You can use the sample app in Oracle JDeveloper to review, deploy, and test against Oracle E-Business Suite. **Important:** To use the sample app, ensure that you apply required sample app patches as described in Applying Prerequisite Patches on the	• `EBSSample.zip`

Folder Name	Subfolder and File Name
Oracle E-Business Suite Server, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2* for your Oracle E-Business Suite instance. Additionally, the REST services used by the sample app must be deployed. For more information, see: • Implementing Oracle E-Business Suite REST Services, page 7-4 • Using the Sample App as a Reference, page B-1.	

Getting Started with Mobile Application Project

Oracle Mobile Application Framework development knowledge is required for developing custom mobile apps for Oracle E-Business Suite. Refer to the Oracle Mobile Application Framework 2.6.3 Migration Notes (https://www.oracle.com/application-development/technologies/maf/maf263migration.html).

Note: For mobile apps built with Oracle E-Business Suite Mobile Foundation Release 9.0, refer to Oracle Mobile Application Framework 2.6.2 Migration Notes (https://www.oracle.com/application-development/technologies/maf/maf262migration.html). For mobile apps built with Oracle E-Business Suite Mobile Foundation Release 8.0, refer to the Oracle Mobile Application Framework 2.5.0 documentation (https://docs.oracle.com/middleware/maf250/mobile/index.html). For mobile apps built with Oracle E-Business Suite Mobile Foundation 7.0, see Oracle Mobile Application Framework 2.4.0 documentation (https://docs.oracle.com/middleware/maf240/mobile/index.html).

Additional Information: If you are upgrading your mobile apps to Oracle E-Business Suite Mobile Foundation Release 9.1 from Release 4.1 or later, perform the needed tasks as described in Migrating Your Mobile Apps, page 5-46.

You can start your mobile application project in either of the following ways:

Option 1: Using Mobile Application Template (Recommended)

This option uses the mobile application template (`EBSMobileAppTemplate.zip`) in which the Login component is already set up.

To use the template, unzip the `EBSMobileAppTemplate.zip` file and open the project file in Oracle JDeveloper. Since the template is already preconfigured with the Login component, in this situation, proceed to Step 3: Setting Up the Sign In Screen, page 5-31 directly (skip the first two steps) and then perform the rest of setup tasks described in this chapter.

Option 2: Creating a New MAF Application

Without using the mobile application template, you need to create a new MAF application from scratch in Oracle JDeveloper and then set up the Login component yourself.

Once the MAF application is created, Oracle JDeveloper creates required folders and other artifacts under the workspace folder. You need to perform the following tasks to set up the Login component for your project first and then complete the rest of the tasks described in this chapter:

- Step 1: Copying the Login Component Files (Conditional), page 5-18
- Step 2: Setting Up the Login Component Libraries (Conditional), page 5-21

Step 1: Copying the Login Component Files (Conditional)

When creating a mobile app from scratch without using the mobile application template, you need to create a new MAF application in Oracle JDeveloper and then set up the Login component by copying the Login component files and then including them in your project.

> **Important:** If the mobile application template is used while creating mobile apps, since the template is already preconfigured with the Login component, skip this step and directly proceed to the Step 3: Setting Up the Sign In Screen, page 5-31.

Perform the following steps to copy the Login component files to your newly created MAF application project:

1. Copy MAF artifacts.

   After the MAF application project is created, the MAF artifacts are automatically generated. Copy these files from the Login component package that contains Oracle E-Business Suite specific content and them replace existing files in your Oracle JDeveloper project.

	Package Directory	Filename	Destination Directory
1	maf/project/adf/META-INF	wsm-assembly.xml	<ApplicationRootFolder>/.adf/META-INF
2	maf/project/adf/META-INF	connections.xml	<ApplicationRootFolder>/.adf/META-INF
3	maf/project/adf/META-INF	maf-config.xml	<ApplicationRootFolder>/.adf/META-INF
4	maf/project/src/META-INF	logging.properties	<ApplicationRootFolder>/src/META-INF
5	maf/project/src/META-INF	maf.properties	<ApplicationRootFolder>/src/META-INF
6	maf/project/ApplicationController/src/META-INF	maf-skins.xml	<ApplicationRootFolder>/ApplicationController/src/META-INF

2. Copy the Login component libraries.

   The Login component contains the following two libraries:

   - `EBSLoginCC.jar`- a MAF feature archive that contains MAF artifacts, such as pages, task flows, and other MAF specific content required for the Login component flows.
   - `EBSLoginLib.jar` - a Java archive file that contains all Java classes implemented in the Login component.

   Copy the Login component libraries listed in the table to a given destination folder. If the destination folder is not present, you need to create one.

	Package Directory	Filename	Destination Directory
1	maf/project/lib	EBSLoginCC.jar	<ApplicationRootFolder>/lib
2	maf/project/lib	EBSLoginLib.jar	<ApplicationRootFolder>/lib

3. Copy Oracle E-Business Suite artifacts.

   Following files are Oracle E-Business Suite specific implementations required by the Login component. Copy these files to your Oracle JDeveloper project. If the destination folders listed here are not present, you need to create them.

	Package Directory	Filename	Destination Directory
1	maf/project/adf/META-INF/ebs	ebs.properties	<ApplicationRootFolder>/.adf/META-INF/ebs
2	maf/project/adf/META-INF/ebs	Custom-EULA-Generic.html	<ApplicationRootFolder>/.adf/META-INF/ebs
3	maf/project/adf/META-INF/ebs	CorporateLogo.png	<ApplicationRootFolder>/.adf/META-INF/ebs
4	maf/project/ApplicationController/public_html/resources/html	ebs-Login.html	<ApplicationRootFolder>/ApplicationController/public_html/resources/html
5	maf/project/ApplicationController/public_html/resources/images	CorporateLogo.png	<ApplicationRootFolder>/ApplicationController/public_html/resources/images
6	maf/project/ApplicationController/public_html/resources/css	ebs-Login.css	<ApplicationRootFolder>/ApplicationController/public_html/resources/css
7	maf/project/ApplicationController/public_html/resources/css	EBSMobile-1.1.css EBSMobile-1.0.css	<ApplicationRootFolder>/ApplicationController/public_html/resources/css
8	maf/project/ApplicationController/public_html/resources/css	EBSMobile-1.1.Android.css EBSMobile-1.0.Android.css	<ApplicationRootFolder>/ApplicationController/public_html/resources/css
9	maf/project/ApplicationController/public_html/resources/css	EBSMobile-1.1.iOS.css EBSMobile-1.0.iOS.css	<ApplicationRootFolder>/ApplicationController/public_html/resources/css
10	maf/project/ApplicationController/public_html/resources/js	ebs-Login.js	<ApplicationRootFolder>/ApplicationController/public_html/resources/js

	Package Directory	Filename	Destination Directory
11	maf/project/ApplicationController/public_html/resources/js	ebs-LoginBundle.* (All files)	<ApplicationRootFolder>/ApplicationController/public_html/resources/js
12	maf/project/ViewController/public_html/resources/js	ebs-ControlActions.js	<ApplicationRootFolder>/ViewController/public_html/resources/js
13	maf/project/ViewController/public_html/resources/images	springboard_list.png	<ApplicationRootFolder>/ViewController/public_html/resources/images
14	This copy step is completed after the app image is prepared, as described in Step 6: Integrating with the Springboard, page 5-36.	android_app_header_icon.png	<ApplicationRootFolder>/ViewController/public_html/resources/images

4. Copy Cordova plug-ins.

   Copy the `maf/project/src/Cordova-Plugin-InAppBrowser` folder to `<ApplicationRootFolder>/src/`.

Step 2: Setting Up the Login Component Libraries (Conditional)

Similar to copying the component files, this step is required only if you do not use the mobile application template to create your mobile apps. If the template is used, skip this step and proceed to Step 3: Setting Up the Sign In Screen, page 5-31.

After copying the Login component files to your project, perform the following tasks to include them in your MAF project:

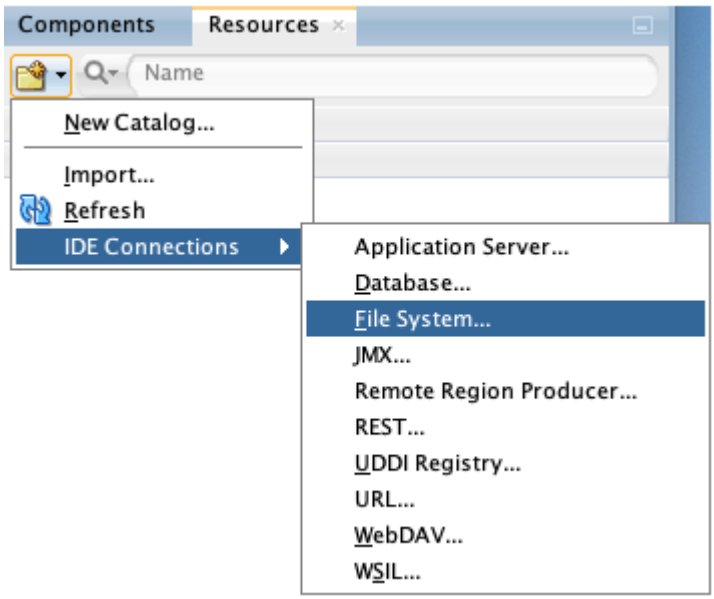
> **Important:** The Login component is configurable by using the `ebs.properties` file to utilize its features. Refer to Supported Properties for the Login Component, page C-1.

1. Create the file system connection.

   1. In Oracle JDeveloper, select Window and click **Resources** to open the Resources window.

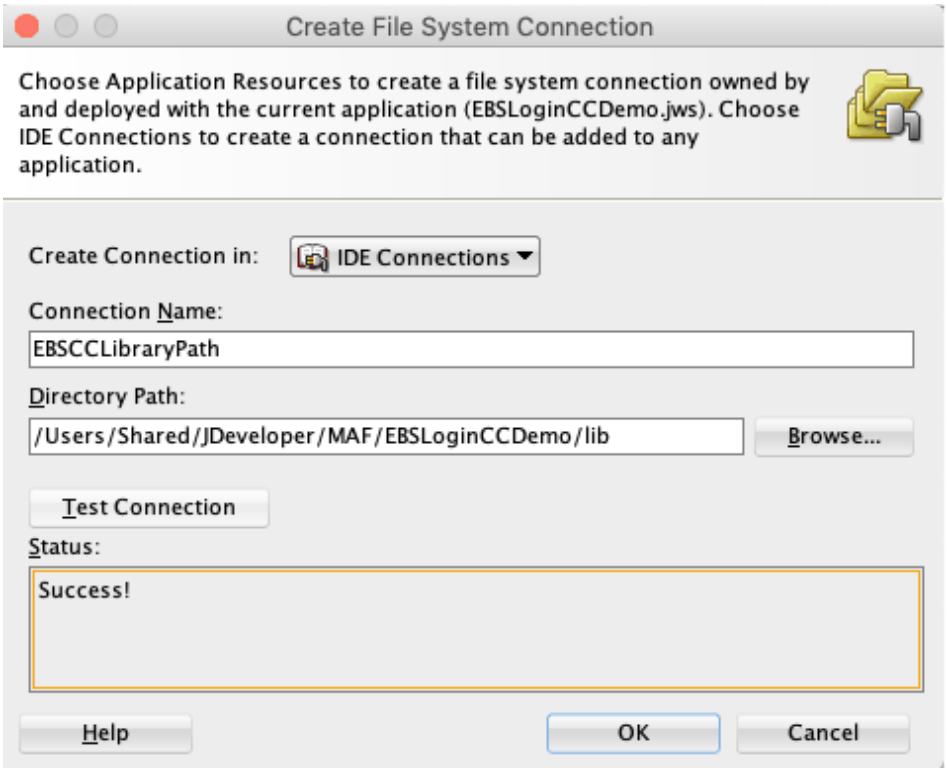
2. Click the folder icon with the + sign, then **IDE Connections**, and then click **File System**.

Oracle JDeveloper Navigation Menu with File System Selected



3. In the Create File System Connection window, enter the following information:

Create File System Connection Window



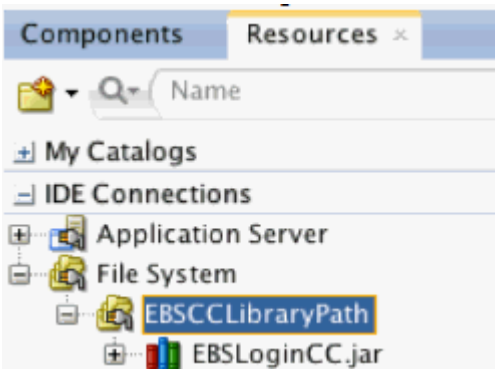
- Connection Name: Enter the connection name, such as "EBSCCLibraryPath".

- Directory Path: Enter the directory path information that points to the `lib` folder under the application's root where you have copied the `EBSLoginCC.jar` file.

  The directory path should point only up to the `lib` folder.

4. In the Resources window, select the IDE Connections panel. Expand **File System** and then **EBSCCLibraryPath** to locate the `EBSLoginCC.jar` file.

Resources Window with the EBSLoginCC.jar File Selected



5. Save your work.

2. Include the Login component archive as an application library.

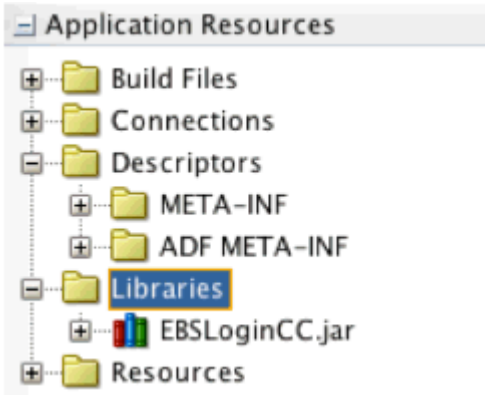
    1. In Oracle JDeveloper, select your MAF application under the Applications window.

    2. Right click the `EBSLoginCC.jar` file in the Resources window.

    3. Add it to Application and select Application Library.

    4. Confirm that `EBSLoginCC.jar` is added as an application library by checking the **Application Resources** panel and then expanding the **Libraries** folder.

Application Resources Window with EBSLoginCC.jar in the Libraries Folder



5. Save your work.

3. Include the Login component library in the ApplicationController project's

Classpath.

1. In Oracle JDeveloper, right click the "ApplicationController" project.

2. Select **Project Properties** from the menu. The Project Properties dialog box appears.

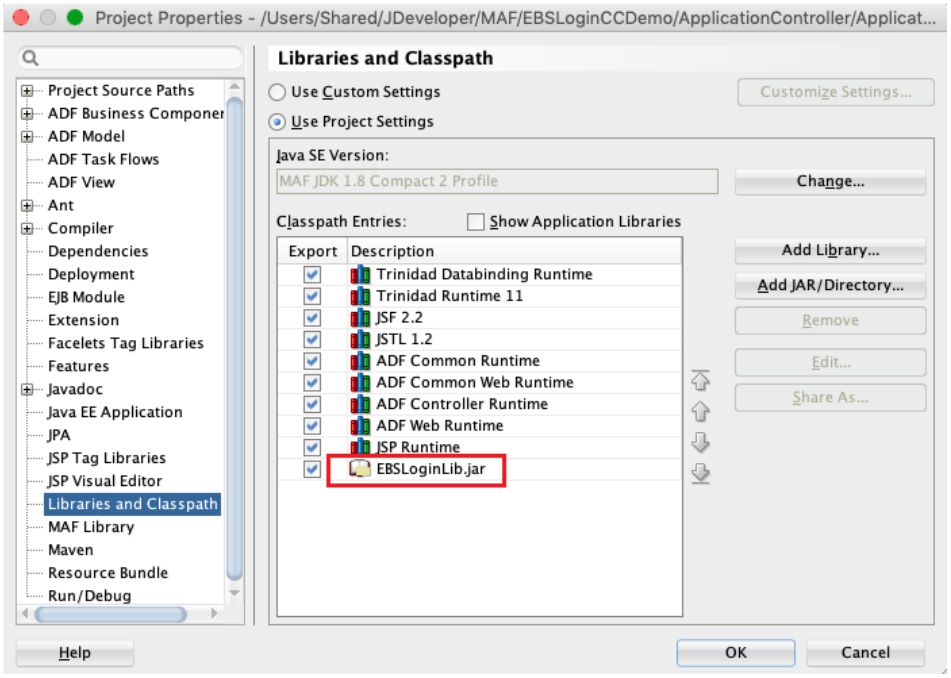
   Select **Libraries and Classpath** and then click **Add JAR/Directory**.

   The Add Archive or Directory dialog box appears.

3. Locate the `EBSLoginLib.jar` folder from the application's lib folder and click **Select** to add the selected `EBSLoginLib.jar` folder.

4. Confirm that the `EBSLoginLib.jar` folder is included and click **OK**.

Libraries and Classpath Window with EBSLoginLib.jar Highlighted in the ApplicationController Project's Classpath



5. Save your work.

4. Include the Login component library in the ViewController project's Classpath.

   1. In Oracle JDeveloper, right click the "ViewController" project.

   2. Select **Project Properties** from the menu. The Project Properties dialog box appears.

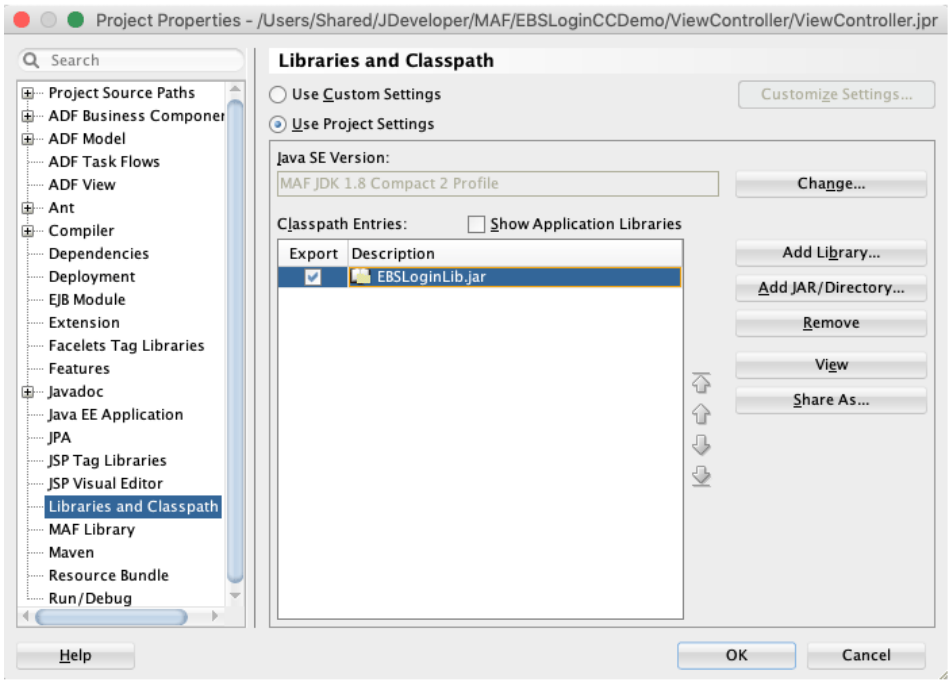
Select **Libraries and Classpath** and then click **Add JAR/Directory**.

The Add Archive or Directory dialog box appears.

3. Locate the `EBSLoginLib.jar` folder from the application's lib folder and click **Select** to add the selected `EBSLoginLib.jar` folder.

4. Confirm that the `EBSLoginLib.jar` folder is included and click **OK**.

Libraries and Classpath Window with EBSLoginLib.jar in the ViewController Project's Classpath



5. Save your work.

5. Include the Login component feature archive into your app's **`maf-application.xml`**. All these features are required in order for the Login component to work correctly.

1. In Oracle JDeveloper, select your mobile application project.

2. In the Applications Navigator, expand the **Application Resources** panel, then the **Descriptors** folder, and then the **ADF META-INF** folder.

   Double click to open the **`maf-application.xml`** file.

3. In the overview editor of the **`maf-application.xml`** file, select **Feature References** from the left panel.

4. Add the following features in the order of sequence listed here:
    1. `oracle.apps.fnd.mobile.login.EULA`
    2. `oracle.apps.fnd.mobile.login.Config`
    3. `oracle.apps.fnd.mobile.login.ConfigSecured`
    4. `oracle.apps.fnd.mobile.login.ResponsibilityPicker`
    5. `oracle.apps.fnd.mobile.login.Springboard`
    6. `oracle.apps.fnd.mobile.login.About`
    7. `oracle.apps.fnd.mobile.login.Signout`
    8. `oracle.apps.fnd.mobile.login.ConnectionDetails`
    9. `oracle.apps.fnd.mobile.login.Diagnostics`

    Important: For the application features you will be creating later for your app, include those features after the Login component's default features.

5. Make sure "Show on Springboard" is set to "true" and "Show on Navigation Bar" is set to "false" only for `oracle.apps.fnd.mobile.login.About`.

    For all other features, ensure that "Show on Navigation Bar" and "Show on Springboard" are both set to "false".

6. Save your work.

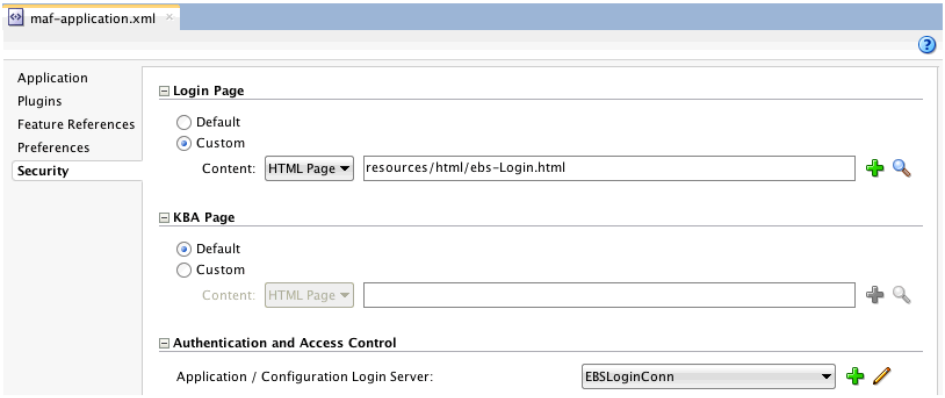
6. Configure your application with Oracle E-Business Suite login page and server.
    1. In Oracle JDeveloper, select your mobile application project.
    2. In the Applications Navigator, expand the **Application Resources** panel, then the **Descriptors** folder, and then the **ADF META-INF** folder.
    3. Double click to open the **`maf-application.xml`** file.
    4. In the overview editor of the **`maf-application.xml`** file, select **Security** from the left panel.
    5. In the Login Page region, enter the following information:
        - Select the "Custom" button.
        - In the Content field, select "HTML Page" from the drop-down list and the

`ebs-Login.html` file for the HTML Page.

The `ebs-Login.html` file should be located under your application's `ApplicationController/public_html/resources/html` folder.

6. In the Application and Access Control region, select "EBSLoginConn" from the drop-down list for the Application / Configuration Login Server field.

Security Tab: Login Page Region, KBA Page Region, Application and Access Control Region



7. Save your work.

7. Perform the following tasks to configure your app with Springboard:

1. In Oracle JDeveloper, select your mobile application project.

2. In the Application Navigator, expand the **Application Resources** panel, then the **Descriptors** folder, and then the **ADF META-INF** folder.

3. Double click to open the **`maf-application.xml`** file.

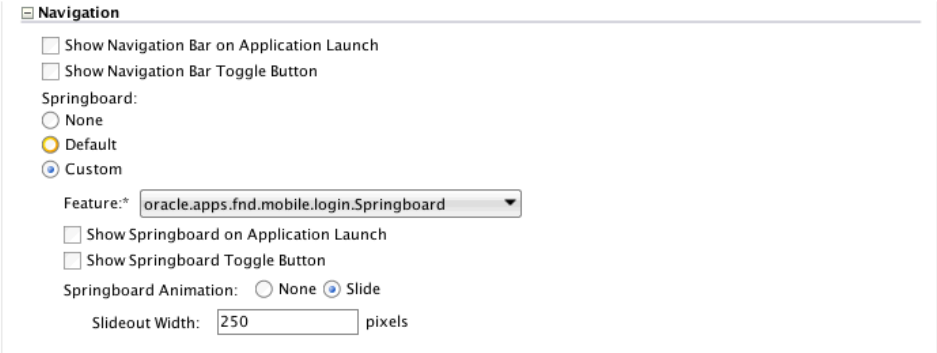
4. In the overview editor of the **`maf-application.xml`** file, select **Application** from the left panel.

5. In the Navigation section, enter the following information:

   - Leave the following check boxes unchecked:
     - Show Navigation Bar on Application Launch
     - Show Navigation Bar Toggle Button
   - Springboard: Select the "Custom" button as the value.

- Feature: Select `oracle.apps.fnd.mobile.login.Springboard` from the drop-down list as the value.
- Leave the following check boxes unchecked:
  - Show Springboard on Application Launch
  - Show Springboard Toggle Button
- Springboard Animation: Select the "Slide" button.
- Slideout Width: Set it to "250" pixels.

Application Tab: Navigation Region



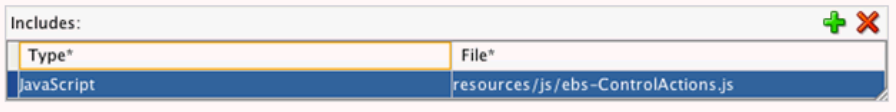
6. Save your work.

8. Perform the following tasks to ensure the Back navigation for Android apps works correctly:

   - Use `ebs-ControlAction.js` to set all the features.

     1. In Oracle JDeveloper, select your mobile application project.
     2. In the Applications Navigator, expand the **ViewController** folder, then the **Application Sources** folder, and then the **META-INF** folder.
     3. Double click **`maf-feature.xml`**.
     4. Select each feature and click the Content tab.
     5. Click the **+** icon for the **Includes** region and enter the following values:

Content Tab: Includes Region



- Type: JavaScript
- File: `resources/js/ebs-ControlActions.js`

6. Repeat this procedure for all the features you include from the Login component as well as the new features you create for your app.
7. Save your work.

9. Add managed beans by performing the following tasks:
   1. In Oracle JDeveloper, select your mobile application project.
   2. In the Applications Navigator, expand the **ViewController** panel, and then the **Web Content** folder.

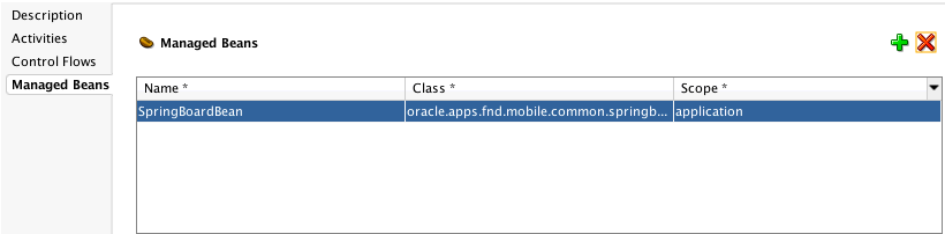
      Note: The **Web Content** folder will not be created until a feature is added in the **`maf-feature.xml`** file. If you do not see the **Web Content** folder under your View Controller project, make sure that you have at least one feature with valid Content added in the **`maf-feature.xml`** file.

      Double click to open the **`adfc-mobile-config.xml`** file.
   3. Select Overview and then click "Managed Beans" from the left panel.
   4. Add the "SpringBoardBean" managed bean with the following information:

Name	Class	Scope
SpringBoardBean	oracle.apps.fnd.mobile.common.springboard.SpringBoardBean	application

      Please note that the managed bean classes come from `EBSLoginCC.jar`.

Managed Beans Tab: Managed Beans Region



5. Save your work.

Step 3: Setting Up the Sign In Screen

The `ebs-LoginBundle.js` file that you copied to your MAF project earlier contains translated Application Names for some of the common Oracle E-Business Suite mobile apps.

In order to show your app name in the Sign In screen, perform the following tasks:

1. In Oracle JDeveloper, select your mobile application project.
2. In the Applications Navigator, expand the **ApplicationController** folder, then the **Web Content** folder, then the **resources** folder, and then the **js** folder.
3. Double click the `ebs-LoginBundle.js` file.
4. Add an entry for your app name. For example, `'APP_TITLE_XXAPPNAME' : 'My App',`.

   Ensure that there is a comma at the end of the entry if it is not the last entry.
5. If your mobile app is translated to different languages, you may add the same entry in all other `ebs-LoginBundle_<langcode>.js` files as well.
6. Double click the `ebs-Login.js` file.

   This JavaScript loads translatable app names on the Sign In screen from `ebs-LoginBundle.js`. You need to change this `ebs-Login.js` file to reference the translation key from `ebs-LoginBundle.js` that corresponds to your app, such as `APP_TITLE_XXAPPNAME`.

   For example, for the Approvals app, string `APP_TITLE_APPROVALS` is referenced in the following line of code:

   ```
   document.getElementById('message').innerHTML =
   getLocaleMessage('APP_TITLE_XXAPPNAME');
   ```

Step 4: Implementing Cordova InAppBrowser Plug-in

The Login component internally uses the Cordova InAppBrowser plug-in to open external URLs within an app. Specifically, implementing this plug-in allows an app to:

- Display Oracle's privacy policy link in the About screen or other links in the End User License Agreement (EULA) screen
- Display external URLs within an app

Register the plug-in by performing the following steps:

1. Copy the folder from `maf/project/src/Cordova-Plugin-InAppBrowser` to `<ApplicationFolder>/src`.

   Note: If you use the Cordova barcode scanner plug-in for your app, move the barcode scanner plug-in as well to the `src` folder and re-register it. This helps you organize all the Cordova plug-ins into the same directory.

2. In Oracle JDeveloper. In the Applications Navigator, expand the **Application Resources** panel, then the **Descriptors** folder, and then the **ADF META-INF** folder.

   Double-click the **`maf-application.xml`** file.

3. In the overview editor that appears, click the Plugins navigation tab.
4. Go to the Additional Plugins section.
5. Click the **+** icon. Navigate to the `src` folder and then select folder "Cordova-Plugin-InAppBrowser".

   This automatically adds an entry for InAppBrowser.

Additionally, after the registration, perform the following tasks if any of your app pages shows URLs to external websites (open outside MAF's default WebView):

1. Go to `maf-feature.xml` and include `ebs-ControlActions.js` to the feature that you want to open an external URL within an app.
2. Invoke the Oracle E-Business Suite Mobile Foundation API `openURLInAppBrowser` in class `oracle.apps.fnd.mobile.common.utils.AppsUtil` from your managed bean or data control method as follows:
   - Use `amx:commandLink` to display external URLs to open outside MAF's WebView, such as:

     ```
     amx:commandLink text="#{row.fileUrl}" shortDesc="#{common.
     CC_ATCH_URLATTACHMENT}" actionListener="#{bindings.openURL.
     execute}"/>
     ```

- In the managed bean or data control method invoke the following Java API with Feature Id and the target URL as input arguments to open the URL outside WebView:

  ```
  AppsUtil.openURLInAppBrowser(featureId, url);
  ```

Step 5: Setting Up Context Initialization

Responsibility picker and organization picker are part of the context initialization flow for those mobile apps that require the security context to fetch data from Oracle E-Business Suite. While some mobile apps may only require responsibility picker, some mobile apps may require a responsibility followed by an organization before the security context can be initialized.

The Login component provides the context initialization flow only up to the responsibility picker. You can insert any number of steps, after the responsibility picker, into the context initialization flow, such as organization picker followed by department picker and so on before the user is directed to the app's landing page.

You can turn on the responsibility picker for the mobile app using the property `oracle.ebs.login.useresppicker=Y` in the `ebs.properties` file. Once the responsibility picker is turned on, based on your app's requirement, you have the following choices:

> **Important:** Oracle E-Business Suite mobile sample app implements organization picker that follows responsibility picker. Developers can use the sample app as a reference.
>
> Context initialization flow is invoked only once when an app is launched for the first time. The selected context parameters, such as responsibility and organization, are saved locally and used to initialize context for every subsequent visit. If a user wants to change the settings, go to the app's Settings screen from Springboard.

- **Use Only the Responsibility Picker in the Context Initialization Flow**

  In this option, the context initialization flow completes after a user selects a responsibility. The required setup tasks are as follows:

  1. Set the property `oracle.ebs.login.responsibilitypicker.usedonebutton=Y` to show **Done** on the Responsibility Picker screen.

  2. After a user selects a responsibility, the user taps **Done** to initialize the selected context and then move to the app's default landing page.

  3. Since context initialization stops at the responsibility picker, set the following properties as suggested here:

- `oracle.ebs.login.init.defaultfeature=`

  This indicates that there are no other steps in the context initialization flow.

- `oracle.ebs.login.initializeresp=Y`

  This indicates the responsibility context is initialized by the Login component after the user taps **Done** on the Responsibility Picker screen.

- `oracle.ebs.login.app.defaultfeature=<App's Default Feature>`

  This indicates the user should be taken to the specified landing page after tapping **Done** on the Responsibility Picker screen.

- **Use Additional Steps in the Context Initialization Flow**

  In this option, the context initialization flow continues after a user selects a responsibility. The required setup tasks are as follows:

  1. Set the property `oracle.ebs.login.responsibilitypicker.usedonebutton=N` to show **Next** on the Responsibility Picker screen.

  2. Since context initialization continues after the responsibility picker, set the following properties as suggested here:

     - `oracle.ebs.login.init.defaultfeature=<App's Org Picker Feature>`

       This indicates the user should be taken to another feature, such as organization picker to continue with the context initialization.

     - `oracle.ebs.login.initializeresp=N`

       This indicates the Login component will not initialize the context, but the app will initialize the context after the context initialization flow completes. The selected responsibility is passed to the app through the application scope variables that can be read using the API given in the next step 3.

     - `oracle.ebs.login.app.defaultfeature=<App's Default Feature>`

       This indicates the user should be taken to the specified landing page after tapping **Done** on the Responsibility Picker screen.

  3. App-specific organization picker feature can read the selected responsibility using the API, `In class oracle.apps.fnd.mobile.common.beans.ResponsibilityAppScopeBean`:

     ```
     ResponsibilityBean respBean = ResponsibilityAppScopeBean.
     getCurrentResponsibilityBean();
     String respId = respBean.getResponsibilityId();
     String appId = respBean.getApplicationId();
     String respDName = respBean.getDisplayName();
     ```

4. The selected responsibility information could be used by the subsequent steps in the context initialization flow to retrieve further information from the Oracle E-Business Suite server.

5. After the app-specific steps in the context initialization flow is completed and the application code is ready to initialize the context, use the following API in `oracle.apps.fnd.mobile.common.RespUtils` class to initialize context and also store the context values in the local database.

   ```
   public static void initializeContext(String username, String
   respId, String applId, String orgId, String securityGroupId,
   String respDisplayName)
   ```

   - `username` - Current user name obtained using `EL expression #{securityContext.userName}`.
   - `respId` - Selected responsibility Id as obtained earlier from `ResponsibilityBean`.
   - `appId` - Selected responsibility's application Id as obtained from `ResponsibilityBean`.
   - `orgId` - If you use your own organization picker, pass the organization Id as selected by a user.
   - `securityGroupId` - Security group Id if applicable.
   - `respDisplayName` - Display name of the responsibility as obtained from `ResponsibilityBean`. This value is stored in the local database to show in the Settings screen later. This value is not sent to the Oracle E-Business Suite server.

6. If you do not want to store the context values in the local database again (assuming it is already stored using above API `initializeContext`), but just reinitialize the context alone with the same values, use the following API in `oracle.apps.fnd.mobile.common.RespUtils` class.

   ```
   public static void reInitializeContext()
   ```

   This is typically the case when you want to reinitialize the application's context in each of your app's feature lifecycle listener class to make sure that before you app feature is loaded, the context is guaranteed to be initialized on the server.

7. For developing custom apps built with Oracle E-Business Suite Mobile Foundation Release 5.0 and onwards

   If you are reinitializing the context in every feature's lifecycle listener class, you can avoid duplicate calls using the following check:

```
if (!AppStateTracker.isContextIntialized()) {
   try {
      RespUtils.reInitializeContext();
   } catch (Exception e) {}
}
```

Step 6: Integrating with the Springboard

This section describes the instructions on how to show the Springboard toggle icon in the landing pages of all the features accessed from Springboard.

Starting from Oracle E-Business Suite Mobile Foundation 5.0, both iOS and Android apps display the same Springboard toggle icon on the left side of the page header. Therefore, use the following steps to modify the landing pages to show the Springboard toggle icon consistently for both iOS and Android platforms:

1. No need to add additional styling to `<amx:panelPage>`. The `<amx:panelPage>` should be like `<amx:panelPage id="pp1">`, where the `id` attribute could have any value.

2. Add the following information under `<amx:panelPage>` from where Springboard should be accessed.

```
<amx:facet name="primary">
  <amx:commandLink actionListener="#{applicationScope.
SpringBoardBean.goToSpringBoard}"
                   styleClass="ebs-sbList-Button" id="cb2"
                   shortDesc="#{EBSCommonBundle.
CC_EBS_SPRINGBOARD}">
   <amx:image id="i3" source="/resources/images/springboard_list.
png"
                         shortDesc="#{EBSCommonBundle.
CC_EBS_SPRINGBOARD}"/>
   <amx:setPropertyListener id="spl3" to="#{pageFlowScope.
disable_id}" from="pglListView"/>
   </amx:commandLink>
</amx:facet>
```

   Please note that `<amx:setPropertyListener>` to `#{pageFlowScope.disable_id)` is to disable the base page controls when Springboard is displayed. The "from" should be the `id` of the PGL that contains the controls to disable.

   Following is an example from the Settings screen.

An Example of the Settings Screen

```
<amx:panelPage id="ppSettings" >
    <amx:facet name="header">
        <amx:outputText value="Settings" id="ot1"/>
    </amx:facet>
    <amx:facet name="primary">
        <amx:commandLink actionListener="#{SpringBoardBean.goToSpringBoard}" styleClass="ebs-sbList-Button"
                         id="cb1" shortDesc="#{commonBundle.CC_EBS_SPRINGBOARD}">
                    <amx:image id="i3" source="/resources/images/springboard_list.png"
                       shortDesc="#{commonBundle.CC_EBS_SPRINGBOARD}"/>
            <amx:setPropertyListener id="spl3" to="#{pageFlowScope.disable_id}" from="pglSettings"/>
        </amx:commandLink>
    </amx:facet>
    <amx:panelGroupLayout id="pglSettings" styleClass="pgl-settings">
```

Step 7: Setting Up Mobile App Access Roles

Oracle E-Business Suite mobile apps use role-based access control to allow users who are assigned the appropriate access roles to access Oracle E-Business Suite. To secure your custom apps, you need to set up required app-specific mobile app access roles first. Mobile applications administrators can then assign these roles to responsibilities that will have access to the custom apps.

For information on creating app-specific access roles in Oracle E-Business Suite, see Creating and Using Mobile App Access Roles, page 3-19.

Update the following properties in the `ebs.properties` file with the role:

- oracle.ebs.login.rolecode=UMX|<role code>
- oracle.ebs.login.roleappname=<app short name>
- oracle.ebs.login.rolemode=parent

For a list of properties supported by `ebs.properties`, see Supported Properties in the Login Component, page C-1.

Step 8: Integrating with the Settings Screen

You can create a MAF feature for the Settings screen that a mobile user can access from Springboard. This may include any app-specific settings that can be captured from the user and used by the app.

App-Specific Preferences

In order for mobile apps to maintain app-specific preferences, the Login component provides PreferenceStore APIs in `class oracle.apps.fnd.mobile.common.utils.PreferenceStore` to store and retrieve preference values in the local database. Some of the use cases of the app-specific settings are:

- Store simple `Name=Value` pairs for a given user. For example, following preferences are specific to the Approvals app:

    - APPROVALS_SORT_ORDER=SENT_DATE
    - APPROVALS_QUICK_APPROVE=Y

- Store `Name=Value[]` for a given user where, for a given user and preference key, you could store an array of values. For example, multiple favorite items can be stored against a single key.
    - APPROVALS_FAVORITE_ITEMS
        - EXPENSES
        - REQUISITIONS
        - PURCHASE_ORDERS

PreferenceStore APIs

The following are common input parameters for the `oracle.apps.fnd.mobile.common.utils.PreferenceStore` APIs.

- **User Name:** User name for whom you want to store the preference value. If the value is null, the current logged-in user's name is used.
- **Module Name:** Use a consistent module name for your app, such as your app internal name. The Login component already uses CCLOGIN as the module name to store its preferences.
- **Preference Name:** Name of the preference you would like to store.
- **Preference Value(s):** Preference value(s) to store. You can either pass a single preference value for a preference name or an array of values.
- **Language:** Use this field to store the language of the preference. It is recommended that you leave this parameter null so that the PreferenceStore APIs will store and retrieve the values based on the device locale. If required, you can pass value to the language parameter for the PreferenceStore APIs in the format "language-territory", such as `en-US`, `ko-KR`, `es-ES`.
    - Language should be a 2-character value based on ISO 639-1 format, such as `en`, `ko`, `es` in lower case.
    - Territory should be a 2-character value based on ISO 3166-1 format, such as `US`, `KR`, `ES` in upper case.
    - If you leave the language parameter null, the PreferenceStore APIs will default the language value to the current device locale.
    - Feature ID: A valid feature ID as configured in the **`maf-feature.xml`** file.

The following table lists the PreferenceStore APIs:

API	Description
`PreferenceStore.getPreference (String userName, String moduleName, String preferenceName, String preferenceValue, String language);`	This API inserts the preference value if not already present. If it is already present, it updates the value.
`PreferenceStore.getPreference (String userName, String moduleName, String preferenceName, String language);`	This API returns the preference value as a string.
`PreferenceStore.setPreferences (String userName, String moduleName, String preferenceName, String preferenceValues[], String language);`	This API inserts an array of preference values if not already present. If the given preference name is already present, it erases existing values and stores the new array of values.
`PreferenceStore.getPreferences (String userName, String moduleName, String preferenceName, String language);`	This API returns an array of preference values as a string.
`PreferenceStore. deleteUserPreference(String userName, String module, String preferenceName);`	This API deletes the preference record from the preference store.
`PreferenceStore.setDefaultFeature (String userName, String featureId);`	This API sets the default feature for the user. This value overrides the default feature configured in the `ebs.properties` file. When the user signs in to the app in subsequent times, the user is navigated to this feature.
`PreferenceStore.getDefaultFeature (String username);`	This AP returns the default feature for the user. This is used internally by the Login component to take the user to the default feature after user signs in to the app.

PreferenceStore APIs Usage

The following are some usage examples of the PreferenceStore APIs:

- Storing language independent values, such as key values

  If you store a preference value that is just a key or a number, such as Sort Order = ASC or DESC, Quick Approvals = YES or NO, Page Size = 20, ensure that these values are stored and retrieved using a constant language.

For example, use language="en-US", or use constant value from oracle.apps.fnd.mobile.common.db.DAOConstants.ENGLISH_LANGUAGE_CODE.

This will guarantee that the same preference value is available across the device locales.

- Storing language dependent values

  If you store a preference value that is translated, it is important to note that if the value is stored in the current device locale. If you try to retrieve it from a different device locale, it will return an empty string.

  If the translated value returns an empty string from the PreferenceStore APIs, it indicates that you should retrieve the value again from the server for the current device locale and persist in the preference store.

  If the app is accessed in multiple locales, each locale will have the language-dependent value in the preference store. In this situation, you could either leave the values for each locale or clean up the existing value using the PreferenceStore.deleteUserPreference API before a new locale value is persisted so that the value is always retrieved fresh from the server.

Default Settings Screen from the Login Component

Apart from any app-specific settings that you can implement, the Login component provides the following features that can also be integrated with app's Settings screen:

- **Responsibility Picker**, page 5-40

  After a user goes through the context initialization flow during the initial launch of the app, the user may have to change the responsibility and organization. In order to do that, the user should go to the Settings screen and change the responsibility and other context parameters. The Login component provides ability to launch the context initialization flow from the Settings screen. This initialization flow is the same flow that the user experienced during the initial launch of the app.

- **Connection Details**, page 5-41

  A mobile user may want to change the server URL for the Oracle E-Business Suite instance to which the app is connected. Additionally, the user or an administrator may want to review the app's configuration parameters that are downloaded from the server. The Login component provides the connection details for this purpose.

- **Diagnostics**, page 5-42

  In order to troubleshoot mobile app related issues, an administrator can work with a mobile app user to enable logging in the Diagnostics screen.

Responsibility Picker (Feature Id - oracle.apps.fnd.mobile.login.ResponsibilityPicker)

When displaying a link for the user to change the current responsibility by relaunching the context initialization flow, it is important to show the current selected responsibility

name in the Settings screen.

Perform the following tasks to integrate the Settings screen with the context initialization flow:

1. Show display name of the current selected responsibility in the Settings screen. The following options are available to get the display name of the current responsibility. The current responsibility name is stored in `#{applicationScope.ResponsibilityAppScopeBean.respDisplayName}`.

    1. Use a backing bean attribute for the outputText field, which will fetch the responsibility display name from the EL.

    2. If the EL is not set, then read the responsibility from the local SQLite database using the following API:

    ```
    PreferenceStore.getPreference(null, DAOConstants.
    APP_USER_PREFERENCES_CCLOGIN_MODULE, ResponsibilityConstants.
    RESPONSIBILITY_DISPLAY_NAME, null);
    ```

2. When the user taps the **Responsibility** link on the Settings screen, perform the following steps:

    1. Reset the feature `oracle.apps.fnd.mobile.login.ResponsibilityPicker`.

    2. Set `#{applicationScope.pageFrom}` to the Settings feature's Id.

    3. Go to feature `oracle.apps.fnd.mobile.login.ResponsibilityPicker`.

3. If you implement a step after the responsibility picker in the context initialization flow, when the user taps **Back** in that step (for example, the Organization Picker screen or any other screen which is present after the Responsibility Picker screen) in the Settings flow, perform the same steps as listed in the previous step 2.

4. Use the following code to navigate to the Responsibility Picker screen:

    ```
    AdfmfContainerUtilities.resetFeature("oracle.apps.fnd.mobile.login.
    ResponsibilityPicker");
    AppsUtil.setELString("#{applicationScope.pageFrom}", "Settings");
    AdfmfContainerUtilities.gotoFeature("oracle.apps.fnd.mobile.login.
    ResponsibilityPicker ");
    ```

Connection Details (Feature Id - `oracle.apps.fnd.mobile.login.ConnectionDetails`)

Perform the following steps to integrate the Settings screen with the Connection Details screen provided from the Login component:

1. Reset the Connection Details feature.

2. Set the application scope variable `#{applicationScope.pageFrom}` to the feature Id from which you are navigating away. For example, if you are navigating

from the Settings screen to the Connection Details screen, set this variable to the feature Id of the Settings feature.

3. Invoke `AdfmfContainerUtilities.gotoFeature` to go to the intended feature.

4. Use the following code to navigate to the Diagnostics screen:

```
AdfmfContainerUtilities.resetFeature("oracle.apps.fnd.mobile.login.
ConnectionDetails");
AppsUtil.setELString("#{applicationScope.pageFrom}", "Settings");
AdfmfContainerUtilities.gotoFeature("oracle.apps.fnd.mobile.login.
ConnectionDetails ");
```

Diagnostics (Feature Id - `oracle.apps.fnd.mobile.login.Diagnostics`)

Perform the following steps to integrate the Settings screen with the Diagnostics screen provided from the Login component:

1. Reset the Diagnostics feature.

2. Set the application scope variable `#{applicationScope.pageFrom}` to the feature Id from which you are navigating away. For example, if you are navigating from the Settings screen to the Diagnostics screen, set this variable to the feature Id of your Settings feature.

3. Invoke `AdfmfContainerUtilities.gotoFeature` to go to the intended feature.

4. Use the following code to navigate to the Diagnostics screen:

```
AdfmfContainerUtilities.resetFeature("oracle.apps.fnd.mobile.login.
Diagnostics");
AppsUtil.setELString("#{applicationScope.pageFrom}", "Settings");
AdfmfContainerUtilities.gotoFeature("oracle.apps.fnd.mobile.login.
Diagnostics ");
```

Local Database Storage

In certain scenarios, you might have to store complex data structures which are not covered by the usage examples for the PreferenceStore APIs discussed earlier. You can create your own tables by leveraging the property `oracle.ebs.login.sql` and pointing to a SQL file that contains the required DDL statements. The Login component during `startup` will run this SQL script to create required tables for your app requirements.

Step 9: Setting Up Default Server URL (Optional)

Starting from Oracle E-Business Suite Mobile Foundation Release 8.0 and onwards, you can preconfigure the Oracle E-Business Suite server URL for your custom apps built with the Login component. Once this task is complete, the custom app users no longer need to enter this URL manually after launching the app.

Depending on whether you use Enterprise Mobility Management (EMM) solutions, you

can configure the default server URL in the following ways:

- If you plan to use Enterprise Mobility Management (EMM) solutions, perform the following tasks:

  1. Configure the following two properties (Server_URL and Server_URL_Allow_Change) in the **`maf-application.xml`** file in Oracle JDeveloper:

     ```
     <adfmf:emmAppConfig>
      <adfmf:property name="Server_URL" type="String" description="EBS
     server URL"/>
      <adfmf:property name="Server_URL_Allow_Change" type="String"
               description="Ability for end user to change the Server
     URL"/>
     </adfmf:emmAppConfig>
     ```

     The Login component already supports the ability to read and use these two configuration properties from EMM. Ensure that you use the same property names when configuring them in **`maf-application.xml`** because the Login component expects to read their values when the app is deployed through an EMM console.

     If these properties are not configured in **`maf-application.xml`**, Oracle E-Business Suite mobile apps installed from an EMM's app catalog will not be able to read and use the values.

  2. Once the apps are deployed to the iOS or Android platform, administrators will host these apps to be installed from EMM and configure the values of these two properties in an EMM console.

     For information about configuring these two properties, see Setup Tasks for Deploying Mobile Apps with Enterprise Mobility Management (EMM) Solutions, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

  For more information on configuring properties for use by EMM solutions, see the "Configuring Properties in MAF Applications for Use by EMM Solutions" section from the "Integrating MAF Applications with EMM Solutions" chapter in *Developing Mobile Applications with Oracle Mobile Application Framework*.

- If you do NOT use Enterprise Mobility Management (EMM) solutions, similar to the apps provided to users through enterprise distribution, you can preconfigure the server URL through the following properties in the `ebs.properties` file:

  - `oracle.ebs.login.server.url`
  - `oracle.ebs.login.server.url.allow_change`

  For more information on setting up these two properties, see Configuring Default Server URL (Optional), page 4-31.

Step 10: Implementing Java Classes

Application Lifecycle Listener

The Login component provides default implementation of Application Lifecycle Listener. If you would like to implement your own Application Lifecycle Listener, perform the following tasks to extend the default Lifecycle Listener implementation to make sure the Login component works correctly.

1. Extend your Application Lifecycle Listener class from the Login component's `oracle.apps.fnd.mobile.common.login.CCLoginLifeCycleListenerImpl`.

2. The first statement in the `start()` method of your Application LifeCycle Listener should be `super.start();`.

3. This statement is extremely important to initialize the SQLite database for your app. If you do not include it, your app will not work as expected.

Step 11: Using Application Logging

If the diagnostics feature is integrated with the app's Settings screen, it can be used to turn on the app logging. The Diagnostics screen allows users to enable or disable the logging feature. The logs written by the app can then be uploaded to the Oracle E-Business Suite server for administrators or developers to review. Log file corresponding to a mobile app is uploaded to the server with the following parameters:

- **Module:** Application's Bundle Id as configured in the `maf-application.xml` file.

- **User Name:** Current signed-in user.

Administrators can retrieve the uploaded log file from the System Administration responsibility using the module name and the user name who uploaded the log file.

Developers can use the AppLogger APIs (`oracle.apps.fnd.mobile.common.utils.AppLogger`) shipped with the Login component to write log statements from app code to client log file when logging is enabled.

Only two log levels are supported:

- **On:** FINE

- **Off:** SEVERE

Please note that developers are required to use only the AppLogger APIs to write client logs. Using MAF's logging interface directly is not allowed.

The following table describes the AppLogger APIs:

API	Description
`logInfo(Class, methodName, message);`	Write informational messages
`logError(Class, methodName, message);`	Write error messages
`logWarning(Class, methodName, message);`	Write warning messages
`logException(Class, methodName, Exception);`	Write exception messages

Step 12: Implementing Corporate Branding

In order for the mobile app to match with your corporate branding, modify the following items that are controlled by the Login component:

- App logo
- App Name
- Splash screen
- End User License Agreement
- Company logo
- Copyright

For step-by-step instructions on modifying the app for corporate branding, see Implementing Corporate Branding, page 9-1.

Step 13: Using the Developer Mode

When deploying and testing the mobile apps that use the Login component, you are required to accept the End User License Agreement (EULA) and then enter the Server URL every time. While testing against a development environment, you should be able to easily bypass these steps and go directly to the Sign In screen.

Use the following properties in `ebs.properties` to enable the developer mode.

- `oracle.ebs.login.development.mode=Y`
- `oracle.ebs.login.development.server.url=<Valid EBS server URL>`
- `oracle.ebs.login.development.logging.finest=Y`

Remember to turn off the developer mode when shipping the app for testing. This way the app can connect to any Oracle E-Business Suite environment during the testing.

Migrating Your Custom Mobile Apps

This section describes information on upgrading your mobile app to Oracle E-Business Suite Mobile Foundation Release 9.1 from Release 4.1 or later. It includes the following tasks:

1. Opening a MAF Application Project, page 5-46
2. Copying Files without Customization, page 5-47
3. Copying Files with Customization, page 5-47
4. (Optional) Enabling Push Notifications (Upgrading to Oracle E-Business Suite Mobile Foundation Release 7.0 or Later Only), page 5-48
5. Deploying Your Apps, page 5-48

Step 1: Opening a MAF Application Project

Before you begin, ensure that you read and truly understand the knowledge of migrating a MAF application, refer to the Oracle Mobile Application Framework 2.6.3 Migration Notes (https://www.oracle.com/application-development/technologies/maf/maf263migration.html).

If you have created a custom app with Oracle E-Business Suite Mobile Foundation Release 9.0, you have used Oracle Mobile Application Framework 2.6.2. In order to migrate your Oracle MAF application from Oracle E-Business Suite Mobile Foundation Release 9.0 to Release 9.1, you need to open your MAF application in Oracle JDeveloper with Oracle Mobile Application Framework 2.6.3. This migrates your application from Oracle Mobile Application Framework 2.6.2 to Oracle Mobile Application Framework 2.6.3.

> **Note:** To migrate your Oracle MAF application from Oracle E-Business Suite Mobile Foundation Release 8.0 to Release 9.0, open your MAF application in Oracle JDeveloper with Oracle Mobile Application Framework 2.6.2 instead. This migrates your application from Oracle Mobile Application Framework 2.5.0 to Oracle Mobile Application Framework 2.6.2. To migrate your Oracle MAF application from Oracle E-Business Suite Mobile Foundation Release 7.0 to Release 8.0, open your MAF application in Oracle JDeveloper with Oracle Mobile Application Framework 2.5.0 instead. This migrates your application from Oracle Mobile Application Framework 2.4.0 to Oracle Mobile Application Framework 2.5.0.

Step 2: Copying Files Without Customization

Once you have migrated the application to Oracle MAF 2.6.3, copy the following files from the Login component package to your newly upgraded MAF application.

1. Copy folder from `maf/project/src/Cordova-Plugin-InAppBrowser` to the `<ApplicationFolder>/src` directory.

   Register the plug-in from the **`maf-application.xml`** file. See: Step 4: Implementing Cordova InAppBrowser Plug-in, page 5-32.

2. Copy the following MAF artifacts.
   - maf-config.xml
   - wsm-assembly.xml
   - maf.properties
   - maf-skins.xml

3. Copy the following the Login component libraries.
   - EBSLoginCC.jar
   - EBSLoginLib.jar

4. Copy the following Oracle E-Business Suite artifacts.
   - EBSMobile-1.1.css
   - EBSMobile-1.1.Android.css
   - EBSMobile-1.1.iOS.css
   - ebs-ControlActions.js

For detailed instructions, see Step 1: Copying the Login Component Files (Conditional), page 5-18.

Step 3: Copying Files with Customization

Before you start copying the customized files, Oracle recommends that you back up existing files as the references of previous customization for your app.

Copy the following files that you could have customized as part of your implementation for your app:

- ebs-Login.js

- ebs-LoginBundle.js
- ebs-LoginBundle_de.js
- ebs-LoginBundle_es_ES.js
- ebs-LoginBundle_es.js
- ebs-LoginBundle_fr_CA.js
- ebs-LoginBundle_fr.js
- ebs-LoginBundle_it.js
- ebs-LoginBundle_ja.js
- ebs-LoginBundle_nl.js
- ebs-LoginBundle_pt_BR.js
- ebs-LoginBundle_zh_CN.js

After copying these files, you need to set up the Login screen, as described in Step 3: Setting Up the Login Screen, page 5-31.

Ensure to redo the customization that you have performed earlier. You can reference the backup files you just created for your previous customization.

Step 4: (Optional) Enabling Push Notifications (Upgrading to Oracle E-Business Suite Mobile Foundation Release 7.0 or Later Only)

After the upgrade and before the deployment, you can optionally enable push notifications, available from Oracle E-Business Suite Mobile Foundation Release 7.0 and onwards, for your upgraded custom apps developed using the Login component.

For information on implementing this feature for your custom apps, see Implementing Push Notifications, page 6-1.

Step 5: Deploying Your Apps

After upgrading your app, you need to redeploy and test the app.

See: Deploying and Testing Mobile Apps, page 5-48.

Deploying and Testing Mobile Apps

After completing the required steps for developing a custom app with the Login component, you can deploy the app for the iOS, Android, or both platforms.

For information on deploying your apps, see Deploying Your Mobile Apps, page 9-39.

You can also use the Login component's developer mode option to quickly deploy and test the app against the development environment.

6

Implementing Push Notifications

Overview

Oracle E-Business Suite provides support for push notifications from Oracle E-Business Suite Mobile Foundation Release 7.0 and onwards. In order for users of the supported mobile apps to receive push notifications on their mobile devices, besides the setup tasks completed by administrators, as described in the Setting Up and Enabling Push Notifications for Oracle E-Business Suite Mobile Apps, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*, you need to perform some tasks both on the Oracle E-Business Suite server and on the mobile client.

- Performing Server-Side Tasks to Send Push Notifications to Mobile Apps, page 6-2

- Performing Client-Side Tasks to Receive Push Notifications in Mobile Apps, page 6-7

Important: Standard Oracle E-Business Suite mobile apps installed directly from the Apple App Store or Google Play do not support push notifications. Push notifications are supported when using Oracle Mobile Hub (OMH) or Oracle Mobile Cloud Service (MCS) in the following apps. Note that in addition to MCS, from Oracle E-Business Suite Mobile Foundation Release 9.0 and onwards, Oracle Mobile Hub provides support for push notifications through Patch 33404902:R12.FND.C for Oracle E-Business Suite 12.2, and Patch 33404902:R12.FND.B for Oracle E-Business Suite 12.1.3.

- Oracle Mobile Approvals for Oracle E-Business Suite, when provided to users through enterprise distribution

- Custom Oracle E-Business Suite mobile apps developed using the Login component from Oracle E-Business Suite Mobile Foundation

See: Using the Login Component to Develop Mobile Apps, page 5-1
.

Push notifications require enterprise distribution due to iOS requirements. The provisioning profile used to build the iOS app and the certificate and private key presented by the Oracle E-Business Suite server to send the push notifications are specific to the iOS app to be used in your organization; therefore, you must obtain your own profile and certificate from the Apple Developer Program. You can then use these to build and deploy the iOS app through enterprise distribution.

- For information on replacing bundle ID or package name in the platform-specific deployment profile, see Modifying an Existing Deployment Profile (Conditional), page 4-23.

- For more information about push notifications, see Understanding Oracle E-Business Suite Mobile Foundation Push Notification System, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

- For more information on enterprise distribution, see Using Mobile Application Archives for Enterprise Distribution, page 4-1.

For information on troubleshooting issues related to the setup or processing of push notifications, see Troubleshooting Tips for Push Notifications, *Oracle E-Business Suite Mobile Apps Developer's Guide, Release 12.1 and 12.2*.

Performing Server-Side Tasks to Send Push Notifications to Mobile Apps

In order for the Oracle E-Business Suite server to send push notifications to the supported mobile apps, you must complete the following tasks:

1. Creating Push Notification Business Events, page 6-2

2. Adding the Push Notification Business Events to the Push Notification System's Event Group, page 6-3

3. Sending Push Notifications to Mobile Devices, page 6-4

Creating Push Notification Business Events

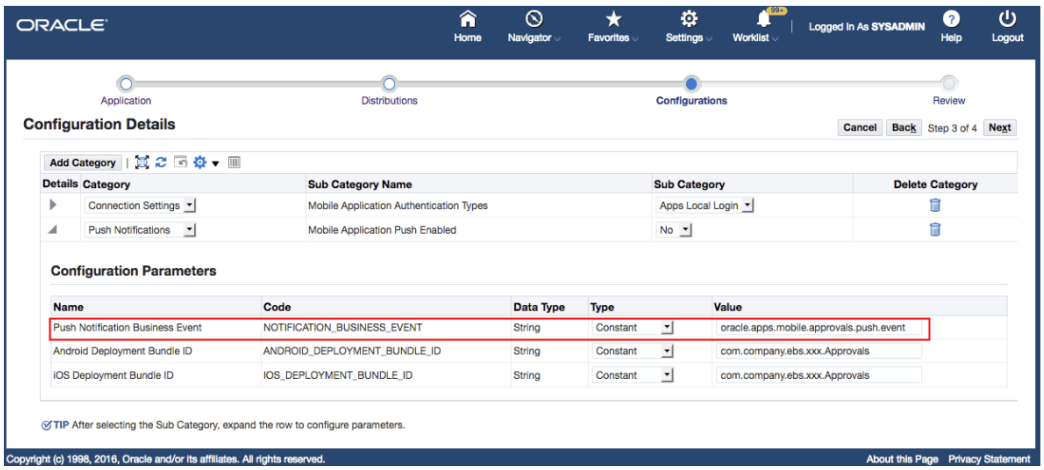
Push notification messages are delivered to mobile devices by Oracle E-Business Suite application code through the use of business events. Each mobile app is associated with its own business event to trigger push notifications. Oracle E-Business Suite application code raises the app-specific business event based on which the Oracle E-Business Suite

Mobile Foundation Push Notification System processes the notifications and delivers them to the mobile devices.

To create the push notification business event for an app, you need to log in to Oracle E-Business Suite as a user who has the **Workflow Administrator Web Applications** responsibility. For information on creating a business event, refer to the *Oracle Workflow Developer Guide*.

The business event you defined in Oracle Workflow will be used as the "Push Notification Business Event" parameter, one of the configuration parameters for the "Push Notifications" category when configuring a mobile app in the Configuration Categories region or registering the application definition metadata in the Configuration Details page through the **Mobile Applications Manager** responsibility.

Configuration Details Page with "Push Notification Business Event" Parameter Highlighted



For example, the enterprise version of the Approvals app (Oracle Mobile Approvals for Oracle E-Business Suite) uses the standard business event `oracle.apps.mobile.approvals.push.event` to trigger push notifications. Note that this event is already seeded and included in the seeded event group for the Push Notification System. You do not need to create the event when enabling push notifications for the enterprise-distributed Approvals app.

Adding the Push Notification Business Events to the Push Notification System's Event Group

After successfully creating the push notification business event corresponding to the supported mobile app, you need to add the newly-created event to the seeded Push Notification System's event group `oracle.apps.mobile.foundation.push.group` as part of the Oracle E-Business Suite Mobile Foundation Push Notification System to which all push notification business events should be added.

> **Note:** This seeded event group already includes the app-specific standard push notification business event, such as `oracle.apps.mobile.approvals.push.event` for Oracle Mobile Approvals for Oracle E-Business Suite.

To add a business event to a group, you need to log in to Oracle E-Business Suite as a user who has the **Workflow Administrator Web Applications** responsibility. For more information, refer to the *Oracle Workflow Developer Guide*.

Sending Push Notifications to Mobile Devices

Oracle E-Business Suite Mobile Foundation Push Notification System provides a PL/SQL API to send push notifications to a given mobile app user. The Push Notification System identifies all the mobile devices that the user has registered to receive push notifications and then sends the same notifications to all registered devices.

For example, if a mobile user `<FIRST_NAME>.<LAST_NAME>` has installed an enterprise-distributed Approval app that is enabled with push notifications on two iOS devices and one Android device. All the three devices are connected to the same Oracle E-Business Suite server. If a push notification is sent to the enterprise-distributed Approvals app for the user, it is delivered to all the three devices by the Push Notification System.

The PL/SQL API signature is as follows:

```
procedure Send (p_event_name in VARCHAR2,
                p_recipient in VARCHAR2,
                p_subject in VARCHAR2,
                p_title in VARCHAR2,
                p_sound in VARCHAR2 default null,
                p_badge in VARCHAR2 default null,
                p_params in wf_parameter_list_t)
```

The following code snippet shows the sample usage of sending push notifications to the enterprise-distributed Approvals app:

```
declare
  -- Replace the text <FIRST_NAME>.<LAST_NAME> with actual recipient of
Push Notification
  l_recipient_role varchar2(320) := '<FIRST_NAME>.<LAST_NAME>';
  -- Replace the number 111111 with actual notification ID
  l_nid number := 111111;

  -- Push Notification event used while registering the App in Mobile
Applications
  -- Manager.
  l_event_name varchar2(100):= 'oracle.apps.mobile.approvals.push.
event';
  l_parameterlist wf_parameter_list_t := wf_parameter_list_t();
  l_subject varchar2(1000);
  l_status varchar2(100);

  l_orig_lang varchar2(64);
  l_orig_terr varchar2(64);
  l_orig_chrs varchar2(64);
  l_orig_date_format varchar2(64);
  l_orig_date_language varchar2(64);
  l_orig_calendar varchar2(64);
  l_orig_numeric_characters varchar2(64);
  l_orig_sort varchar2(64);
  l_orig_currency varchar2(64);
  l_orig_client_timezone varchar2(64);
  l_recip_lang  varchar2(64);
  l_recip_terr varchar2(100);

begin

       select status
       into l_status
       from wf_events
       where name=l_event_name;

      if (l_status <> 'ENABLED') then
        return;
      end if;

      -- Sending translated push notification message
      -- Step 1 - Retain original NLS context
      wf_notification_util.getNLSContext(p_nlsLanguage => l_orig_lang,
                                         p_nlsTerritory => l_orig_terr,
                                         p_nlsCode => l_orig_chrs,
                                         p_nlsDateFormat =>
l_orig_date_format,
                                         p_nlsDateLanguage =>
l_orig_date_language,
                                         p_nlsNumericCharacters =>
l_orig_numeric_characters,
                                         p_nlsSort => l_orig_sort,
                                         p_nlsCalendar =>
l_orig_calendar,
               p_clientTimezone => l_orig_client_timezone);

      -- Find out mobile user's device language and territory preference
      l_recip_lang := fnd_mbl_notification.GetDeviceLang
(l_recipient_role, l_event_name);
      l_recip_terr := fnd_mbl_notification.GetDeviceTerr
(l_recipient_role, l_event_name);

      -- Set NLS context to that of the push notification recipient.
Only  language and
      -- territory are available from device registration, for rest of
the NLS context,leave it to DB default
```

```
wf_notification_util.SetNLSContext(p_nlsLanguage => l_recip_lang,
                                        p_nlsTerritory =>
l_recip_terr);

      -- Get subject in recipient's NLS context
      l_subject := wf_notification.GetSubject(l_nid, 'text/html');

      -- Reset to original NLS context
      wf_notification_util.SetNLSContext(p_nlsLanguage => l_orig_lang,
                                        p_nlsTerritory => l_orig_terr,
                                        p_nlsDateFormat =>
l_orig_date_format,
                                        p_nlsDateLanguage =>
l_orig_date_language,
                                        p_nlsNumericCharacters =>
l_orig_numeric_characters,
                                        p_nlsSort => l_orig_sort,
                                        p_nlsCalendar =>
l_orig_calendar,
               p_clientTimezone => l_orig_client_timezone);

      wf_event.AddParameterToList('WF_NOTIFICATION_ID', to_char(l_nid),
l_parameterlist);
      fnd_mbl_notification.Send(p_event_name => l_event_name,
                                p_recipient => l_recipient_role,
                                p_subject => l_subject,
                                p_title => null,
                                p_params => l_parameterlist);
end;
/
commit;
```

Note: In this sample code, the NLS context is changed to the language of the mobile app user's device locale to translate the message. If the mobile app user's language is not supported by Oracle E-Business Suite, then the message is sent in the base language.

Sending Notification Identifier

When a push notification is sent to a mobile device, in addition to displaying the notification subject shown as the message to the mobile user, you can send internal values regarding that specific notification that can be used by the mobile app code for further processing. For example:

- For the enterprise-distributed Approvals app, when a push notification is sent for an approval request, the Workflow Notification ID (`WF_NOTIFICATION_ID` as shown in the sample code) corresponding to that approval can be sent along with the push notification message. This allows the app to use that ID to fetch the notification details and then in turn allows the user to drill down to the notification details page.

- For an offline mobile app, such as Oracle Mobile Field Service, if data is updated on the server, a signal can be sent through the push notification message to trigger the mobile app to redownload the data from the server.

Performing Client-Side Tasks to Receive Push Notifications in Mobile Apps

In order for Oracle E-Business Suite mobile apps to receive push notifications, you must complete the following tasks on the mobile client:

1. Enabling the Push Plug-in, page 6-7
2. Migrating to Firebase Cloud Messaging (Android Only), page 6-8
3. Handling Push Notifications (Optional), page 6-8

Enabling the Push Plug-in

Push notifications are currently supported when using Oracle Mobile Hub (OMH) from Release 9.0 and onwards with a patch applied or using Oracle Mobile Cloud Service (MCS) for custom mobile apps developed using Oracle E-Business Mobile Foundation as well as for the Approvals app when provided to users through enterprise distribution.

For information about the patch of using Oracle Mobile Hub from Oracle E-Business Suite Mobile Foundation Release 9.0, see Implementing Push Notifications, page 6-1.

To allow these mobile apps to receive push notifications on the mobile devices, you should enable the Push plug-in for the corresponding Mobile Application Framework (MAF) project.

Instructions to Enable the Push Plug-in

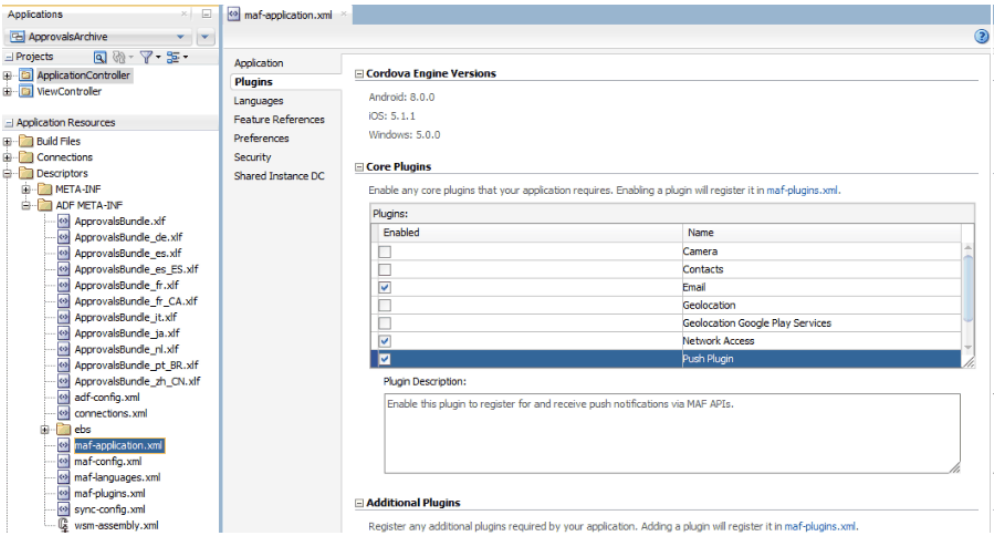
After you have created the MAF project for your mobile app, perform the following tasks to enable the Push plug-in:

1. Open Oracle JDeveloper.
2. In the Applications Navigator, expand the **Application Resources** panel, then the **Descriptors** folder, and then the **ADF META-INF** folder.

   Double-click the **`maf-application.xml`** file to open the overview editor for the **`maf-application.xml`** file.
3. Select the **Plugins** navigation tab.
4. In the Core Plugins region, select the **PushPlugin** check box to enable the Push plug-in.

   Note: The **PushPlugin** check box is unselected by default in the Mobile Application Archive (MAA) file.

Plugins Tab: Core Plugins Region with PushPlugin Selected



5. Save your work.

   This enables the MAF application to first register for push notifications with Oracle E-Business Suite and then receive push notifications.

Migrating to Firebase Cloud Messaging (Android Only)

For push notifications on Android, Oracle E-Business Suite mobile apps use Firebase Cloud Messaging (FCM) to replace Google Cloud Messaging (GCM), which has been deprecated by Google. To migrate to FCM, follow the instructions provided by Google at `https://developers.google.com/cloud-messaging/android/android-migrate-fcm`. After completing the migration or creating a new FCM app, download the generated `google-services.json` file to the root directory of your mobile app (parallel to the `ViewController` and `ApplicationController` directories), as demonstrated by the following example.

```
MAFappRootDirectory
  MAFapp.jws
  ApplicationController
  ViewController
  google-services.json
  ....
```

This completes the migration of your Oracle E-Business Suite mobile apps to use FCM.

Handling Push Notifications (Optional)

When a push notification arrives on a mobile device, the mobile device's user can view the notifications in the notification center. When the user selects the notification from the notification center, the device opens the mobile app corresponding to that

notification. When the mobile app is launched through a selection of a push notification, instead of launching the app in a regular way through the landing page, you can optionally choose to perform some notification handling tasks to let the app directly launch the mobile app page where it displays the appropriate details corresponding to that push notification.

Take an enterprise-distributed Approvals app as an example. A manager receives a push notification "Expense Report W1234 for $500.00 is submitted for approval". When the manager selects that notification from the notification center, the Approvals app displays the Expense Report approval notification details. If you choose not to handle the notifications, the app will be launched normally by taking the user to the regular landing page.

Although handling push notifications is an optional task, from the user experience perspective, it is important for a user to quickly navigate to an appropriate page to obtain the message details when the app is launched through a push notification.

Instructions to Handle Push Notifications

Perform the following tasks to allow your mobile app to handle the push notifications and guide the app users to an appropriate mobile page:

> **Note:** Oracle E-Business Suite mobile apps that implement push notifications using enterprise distribution from the associated MAA files have already completed this optional task. The instructions described here only apply for the custom smartphone apps.

> **Note:** Push notifications cannot be tested in iOS simulator or Android emulator. Instead, you need to install the app on a mobile device to test push notifications.

- **Implementing the PushEventListener Interface**

  The `PushEventListener` interface provides method `onMessage(Event)` that should be implemented by the mobile app to handle push notifications. The push notification message is available in the Event object.

  The following code snippet shows the sample implementation of the enterprise-distributed Approvals app:

```
package oracle.apps.fnd.mobile.approvals;

import oracle.adf.model.datacontrols.device.DeviceManagerFactory;
import oracle.adfmf.framework.api.AdfmfContainerUtilities;
import oracle.adfmf.framework.event.Event;
import oracle.adfmf.json.JSONException;
import oracle.adfmf.json.JSONObject;
import oracle.apps.fnd.mobile.common.notifications.
PushEventListener;
import oracle.apps.fnd.mobile.common.utils.AppLogger;
import oracle.apps.fnd.mobile.common.utils.AppsUtil;

public class ApprovalsPushMsgLsnr implements PushEventListener {

    public ApprovalsPushMsgLsnr() {
        super();
    }

    @Override
    public void onMessage(Event event) {
        JSONObject obj;
        try {
            obj = new JSONObject(event.getPayload());
            String deviceOS = DeviceManagerFactory.
getDeviceManager().getOs();
            String subject = "";

            // Read push notification message
            if ("iOS".equals(deviceOS)){
                subject = obj.getJSONObject("alert").getString
("body");
            }else{
                subject = obj.getString("alert");
            }

            // Read the notification identifier associated to the
push notification
            String notificationID = obj.getJSONObject("data").
getString("WF_NOTIFICATION_ID");

            // App feature to go to when notification is pressed.
Take user to approval details page.
            // Implementation to navigate to a given page can differ
from app to app.
            // Set required input param values to application scope
that can be used for navigation later.
            AppsUtil.setELString("#{applicationScope.
notificationID}", notificationID);
            AppsUtil.setELString("#{applicationScope.subject}",
subject);

            // Navigate to required app feature corresponding to the
notification identifier
            if(event.getApplicationState() != Event.
APPLICATION_STATE_FOREGROUND){
                AdfmfContainerUtilities.resetFeature("oracle.apps.
fnd.mobile.approvals.Approvals", false);
                AdfmfContainerUtilities.gotoFeature("oracle.apps.
fnd.mobile.approvals.Approvals");
            }
        } catch (JSONException e) {
            AppLogger.logError(ApprovalsPushMsgLsnr.class,
"onMessage", AppsUtil.message(e));
        }
    }
```

```
}
```

- **Instantiating the PushEventListener Object**

  Once the `PushEventListener` interface is implemented, instantiate the `PushEventListener` object in your app's lifecycle listener.

```
package oracle.apps.fnd.mobile.approvals;
import oracle.apps.fnd.mobile.common.login.
CCLoginLifeCycleListenerImpl;
import oracle.apps.fnd.mobile.common.notifications.
PushEventListener;
import oracle.apps.fnd.mobile.common.utils.AppLogger;

public class ApprovalsLifeCycleListener extends
CCLoginLifeCycleListenerImpl {

    public ApprovalsLifeCycleListener() {
    }
    public void start() {
        super.start();
    }
    public void stop() {
        // Add code here...
    }
    public void activate() {
        // Add code here...
    }
    public void deactivate() {
        // Add code here...
    }
    public PushEventListener registerPushListener() {
        return new ApprovalsPushMsgLsnr();
    }
}
```

  The application lifecycle listener described above extends the Oracle E-Business Suite Mobile Foundation Login Component's `oracle.apps.fnd.mobile.common.login.CCLoginLifeCycleListenerImpl;` and should be configured in your app's **`maf-application.xml`** file.

7

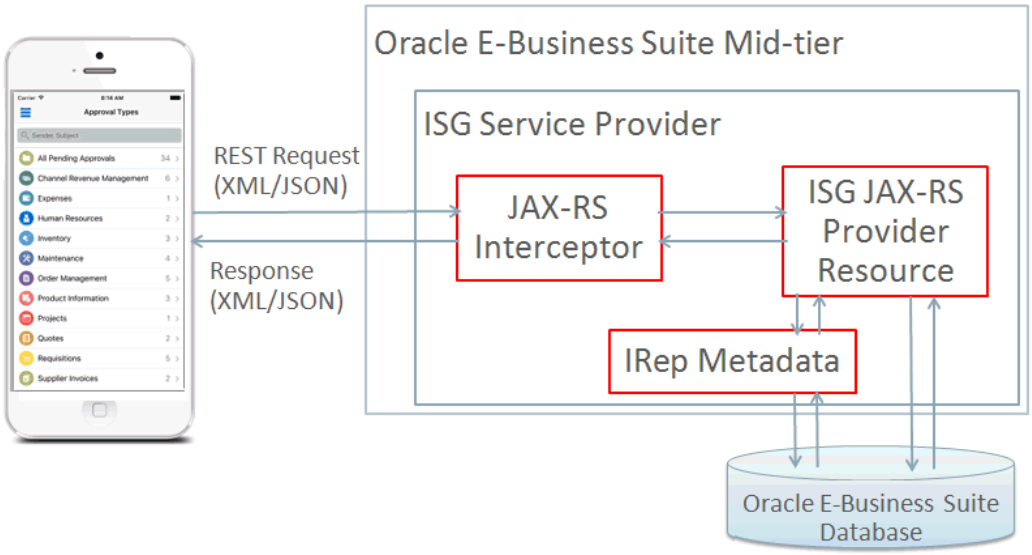
Implementing Oracle E-Business Suite APIs as REST Services for Mobile Apps

Overview

Mobile apps exchange data with Oracle E-Business Suite through REST services provided through Oracle E-Business Suite Integrated SOA Gateway (ISG).

Oracle E-Business Suite Integrated SOA Gateway is an intrinsic part of Oracle E-Business Suite for service enablement. It allows web service clients to make use of the REST services provided from Oracle E-Business Suite. To accomplish this goal, it consists of many essential components. For example, Oracle Integration Repository, a key component that contains numerous interface endpoints exposed by applications throughout Oracle E-Business Suite. A service provider, a component that deploys Oracle E-Business Suite interfaces published in the Integration Repository as REST web services, as shown in the following diagram:

REST Service Provider Architecture



To better understand how Oracle E-Business Suite REST services can be used, the following topics are included in this chapter:

- Understanding Oracle Integration Repository, page 7-2
- Configuring Oracle E-Business Suite REST Services, page 7-3
- Implementing Oracle E-Business Suite REST Services, page 7-4

Understanding Oracle Integration Repository

Oracle Integration Repository is an essential component in Oracle E-Business Suite Integrated SOA Gateway. It is the centralized repository that contains numerous interface endpoints within the Oracle E-Business Suite. These integration interfaces can be built in native technologies or technology products such as PL/SQL, Java, Application Module Services, Concurrent Programs, XML Gateway, and Business Service Objects.

Oracle E-Business Suite Integrated SOA Gateway provides the functionality to expose these integration interfaces published in the Integration Repository as SOAP and REST based web services.

Note: Oracle recommends REST-based services for mobile apps.

Although Oracle E-Business Suite Integrated SOA Gateway is supplied as part of Oracle E-Business Suite, you need to configure Oracle E-Business Suite in order to use its functionality. See: Configuring Oracle E-Business Suite REST Services, page 7-3.

Configuring Oracle E-Business Suite REST Services

To access Oracle E-Business Suite REST services, if you have not configured Oracle E-Business Suite Integrated SOA Gateway (ISG), follow the setup and configuration steps as described in the following documents:

> **Note:** Configuring Oracle E-Business Suite REST services provided through ISG is required only if you use Oracle E-Business Suite REST APIs for custom app development or use a sample app from Oracle E-Business Suite Mobile Foundation Login component. However, it is not required if you use standard Oracle E-Business Suite mobile apps installed from the Apple App Store or Google Play, or apps provided to users through enterprise distribution.
>
> For information on custom app development for Oracle E-Business Suite mobile apps, see Using the Login Component to Develop Mobile Apps, page 5-1.

- For Oracle E-Business Suite 12.2, see "Configuring Oracle E-Business Suite REST Services" section, *Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12.2*, My Oracle Support Knowledge Document 1311068.1.

  > **Note:** Oracle E-Business Suite REST services provided through Oracle E-Business Suite Integrated SOA Gateway do not depend on Oracle SOA Suite and Oracle E-Business Suite Adapter. These REST services are deployed on an Oracle E-Business Suite application server. Therefore, the configuration steps described for REST services in Document 1311068.1 are required to be performed only on Oracle E-Business Suite.

  Ensure that you apply the latest patches for REST services in My Oracle Support Knowledge Document 1311068.1.

- For Oracle E-Business Suite 12.1.3, ensure that you configure Oracle E-Business Suite Integrated SOA Gateway to enable the REST service feature. If Oracle E-Business Suite Integrated SOA Gateway is not configured, follow the setup tasks as described in My Oracle Support Knowledge Document 556540.1 to configure Oracle E-Business Suite Integrated SOA Gateway Release 12.1.3.

  If Oracle E-Business Suite Integrated SOA Gateway is already configured in your instance, then apply the REST service patches in the sequence mentioned in the *Oracle E-Business Suite Integrated SOA Gateway 12.1.3 REST Services Update*, My Oracle Support Knowledge Document 1998019.1 to enable the REST service feature.

Implementing Oracle E-Business Suite REST Services

Once you have completed the required setup tasks on configuring Oracle E-Business Suite REST services, you can start implementing REST services for the mobile apps.

This section describes critical information on managing REST service lifecycle activities including searching and deploying your desired APIs as REST services, viewing deployed services through WADL descriptions, and granting user access privileges for the services. It also includes information on testing and troubleshooting these APIs during custom app development.

- Implementing APIs as Oracle E-Business Suite REST Services, page 7-4
- Testing and Validating the REST Services, page 7-13
- Troubleshooting Tips, page 7-18

Implementing APIs as Oracle E-Business Suite REST Services

You can implement Oracle E-Business Suite REST services through the following ways:

- Using Oracle Published Mobile APIs, page 7-4
- Implementing Custom Interfaces, page 7-12

Using Oracle Published Mobile APIs

Starting from Oracle E-Business Suite Mobile Foundation Release 6.1, Oracle provides APIs corresponding to selected functionality in the Oracle E-Business Suite mobile apps. These APIs are available in the Oracle Integration Repository where you can search, view, and deploy them as REST services for your custom app development.

For a list of the mobile APIs published in the Oracle Integration Repository for each app, see Oracle E-Business Suite Mobile APIs Available in the Oracle Integration Repository, page A-1.

> **Important:** Mobile APIs published in the Oracle Integration Repository have no dependency on Oracle E-Business Suite Mobile Foundation and therefore the availability of mobile app APIs in the repository is not limited to the apps developed based on Oracle E-Business Suite Mobile Foundation. For example, the "Oracle Mobile iProcurement for Oracle E-Business Suite" app is developed based on Oracle E-Business Suite Mobile Foundation, but the APIs published for that app can be used in another custom app that is developed using any other mobile development framework.
>
> Before searching and using these APIs for your custom app

development, you must apply required patches to make these APIs available in your Oracle E-Business Suite instance. For the patch information, see Applying Prerequisite Patches on the Oracle E-Business Suite Server, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Use the following steps to implement these APIs as REST services in Oracle E-Business Suite:

1. Search an Oracle E-Business Suite public interface that provides the desired functionality.

   All Oracle E-Business Suite mobile app APIs are categorized with same Business Entity **Mobile Optimized API**. This allows us to easily locate those APIs published for the mobile apps.

   In addition to the common business entity **Mobile Optimized API** for mobile apps, APIs can be associated with other business entities corresponding to their functional areas. For examples, each API published for "Oracle Mobile Approvals for Oracle E-Business Suite" (or the Approvals app) has two business entities associated with it:

   - **Mobile Optimized API**, the common business entity for all Oracle E-Business Suite mobile APIs
   - **Workflow Notifications**, a workflow related business entity specializing for the workflow area

     Note: You may search or browse interfaces by different categorizations such as product, business entity, or interface type.

     Interface types of PL/SQL APIs, Concurrent Programs, and Java-based APIs including Application Module Services and Java Bean Services can be published as REST services. Currently, all Oracle E-Business Suite mobile APIs are published as "Java" Interface Type and "Application Module Services" as Interface Subtype.

Searching by Business Entity

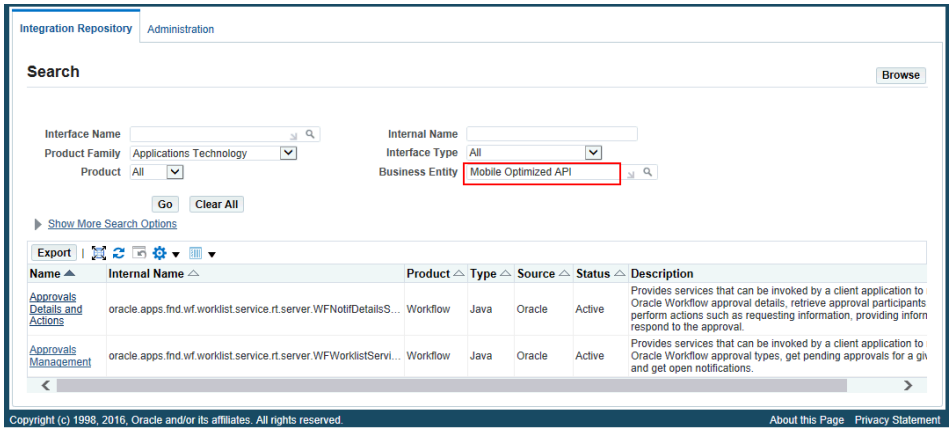
Once you log in to the Oracle Integration Repository through the **Integrated SOA Gateway** responsibility, click **Search**. The Search page appears where you can search and locate your desired mobile app's APIs.

In the Search page, you can search mobile APIs by either of the following business entities:

- Search by the common mobile business entity **Mobile Optimized API** to locate desired mobile apps' APIs in the repository.

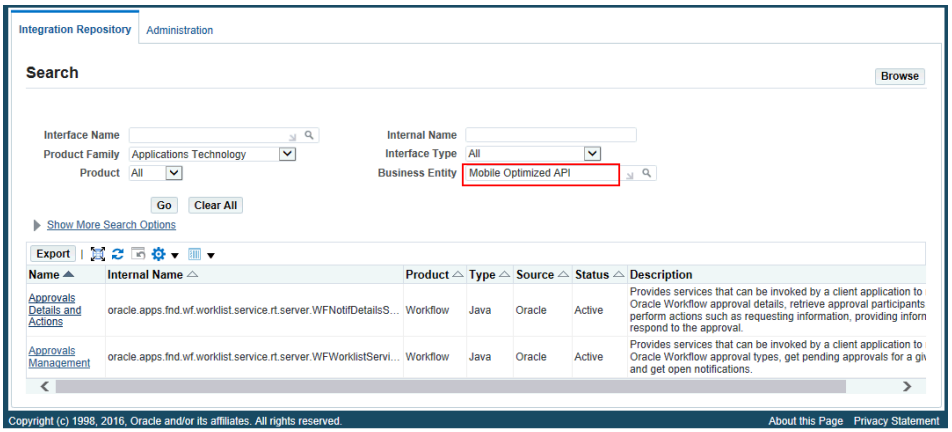
After all required patches for enabling mobile APIs available in Oracle E-Business Suite Mobile Foundation Release 6.1 are applied on your Oracle E-Business Suite instance, you can search by the common mobile business entity **Mobile Optimized API** to list all Oracle E-Business Suite mobile apps' APIs.

Search Page with "Mobile Optimized API" Business Entity Highlighted



To limit the search result for your desired APIs, you can enter additional search values besides the **Mobile Optimized API** business entity. For example, to retrieve the APIs currently available for the Approvals app, you can enter "Applications Technology" as the product family along with the **Mobile Optimized API** business entity to locate the APIs for the Approvals app grouped under the "Applications Technology" product family.

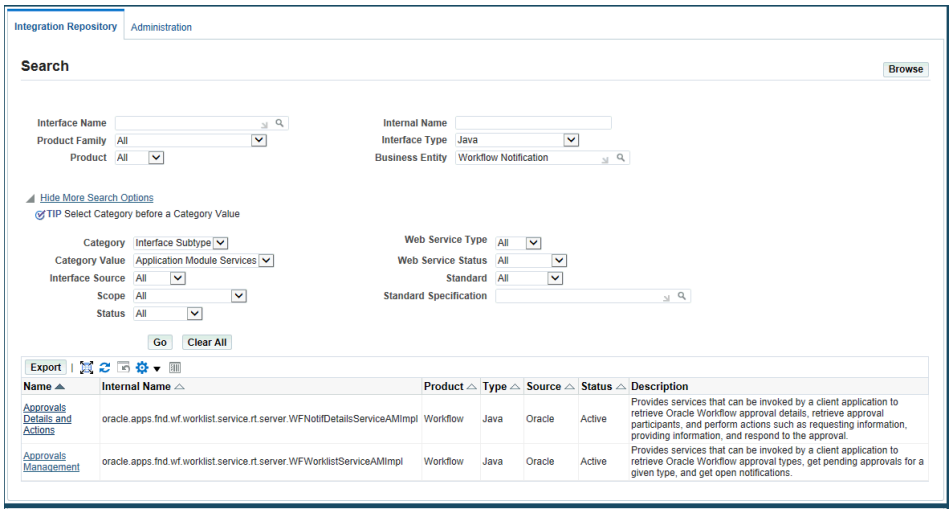
Search Page with "Mobile Optimized API" Business Entity Highlighted and a Specific Product Family as Search Criteria



- Searching by the business entity for a functional area (such as `Workflow Notifications`) along with other search criteria to locate desired APIs.

  All Oracle E-Business Suite mobile APIs available in the repository are published as "Application Module Services", a subtype of Java interface. To locate these APIs corresponding to the Approvals app's functionality, in addition to selecting `Workflow Notifications` as the business entity, you need to specify the 'Java' interface type first and then click the **Show More Search Options** link in the Search page. Select "Interface Subtype" in the Category field and then choose "Application Module Services" in the Category Value field before clicking **Go** to retrieve the search results.

Search Page with Show More Search Options Fields to Search for Approvals App's APIs

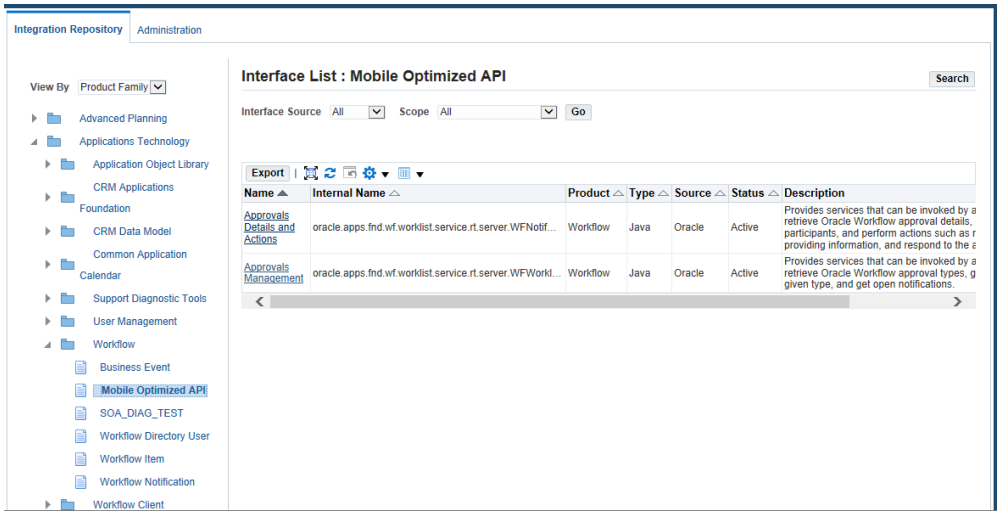


Browsing by Product Family

Alternatively, you can locate desired mobile APIs by browsing the list of interfaces displayed in a tree structure. In fact, the Browse feature appears by default when accessing the Oracle Integration Repository. In the left pane of the browse page, select 'Product Family' in the View By field and then expand the tree nodes to locate your desired **product family**, then **product**, and then **business entity**.

For example, expand the **Applications Technology** product family node, and then the **Workflow** product node from the tree structure. Select the **Mobile Optimized API** business entity node to display the mobile APIs grouped under the Workflow product in the right pane of the page.

Browse Page with View By "Product Family"



2. Create security grants for an interface.

   Oracle E-Business Suite Integrated SOA Gateway utilizes the Oracle E-Business Suite Function Security and Data Security features to ensure that only users with authorized privileges can access certain methods of an interface.

   Note that if your custom apps are developed based on Oracle E-Business Suite Mobile Foundation, the REST APIs consumed by the mobile apps should be granted to the respective mobile app access roles that you will create later during the server-side setup. Oracle E-Business Suite Mobile Foundation uses mobile app access roles to check if a user has the privilege to access the associated mobile apps and then loads relevant responsibilities for that user. See: Creating and Using Mobile App Access Roles, page 3-19.

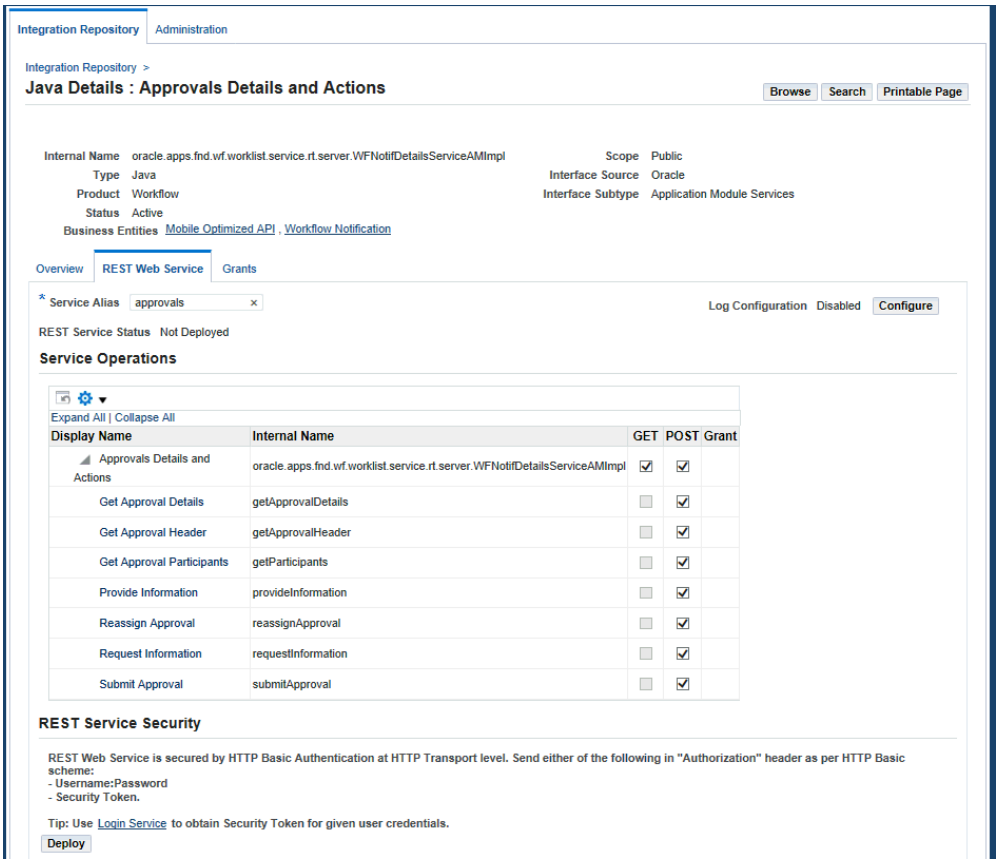
   For information on creating security grants, refer to "Managing Grants for Interfaces with Support for SOAP and REST Web Services" in Administering Native Services chapter of the *Oracle E-Business Suite Integrated SOA Gateway Implementation Guide*.

3. Deploy an interface as a REST Service to an Oracle E-Business Suite application server.

   Once your desired interface is retrieved from a search, click the interface name to view the interface details. If the associated REST service is not available, you can deploy the interface as a REST service by entering the service alias and then clicking **Deploy** in the REST Web Service tab.

   For example, enter "`approvals`" as the service alias name for the "Approvals Details and Actions" interface and then click **Deploy** to deploy the interface as a REST service.

Interface Details Page: REST Web Service Tab with "Not Deployed" REST Service Status



If the REST service is deployed successfully, "Deployed" should appear as the REST Service Status value indicating that the service is available for use on the Oracle E-Business Suite server.

For more information about deploying REST services, refer to Administering Native Services chapter in the *Oracle E-Business Suite Integrated SOA Gateway Implementation Guide*.

4. View the WADL description for a REST service from the Oracle Integration Repository.

   Once a REST service is successfully deployed, you can view the deployed WADL description by clicking the **View WADL** link in the REST Web Service tab.

   - WADL URLs are of the form:

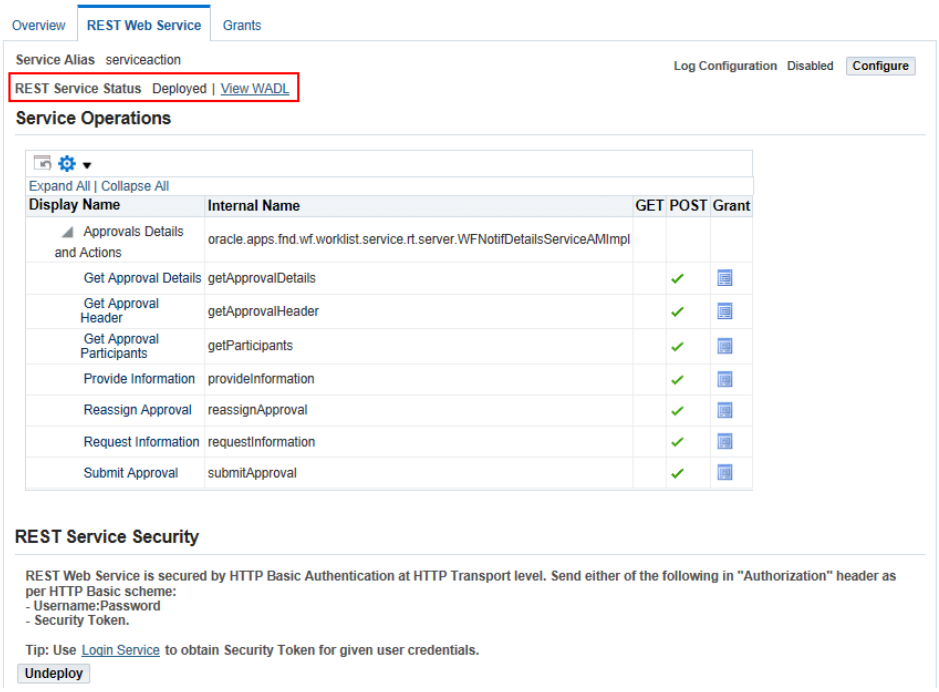
     `http://<hostname>:<port>/webservices/rest/<service_alias>?WADL`

     For example, if `approvals` is used as the service alias for the "Approvals Details and Actions" interface (`oracle.apps.fnd.wf.worlist.service.`

`rt.server.WFNotifDetailsServiceAMImpl`), after you deploy the service, the REST service WADL URL could be like:

`http://<hostname>:<port>/webservices/rest/`**`approvals`**`?WADL`

REST Web Service Tab with "Deployed" REST Service Status and View WADL Link Highlighted



Refer to "Reviewing WADL Element Details" in Discovering and Viewing Integration Interfaces chapter of the *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

- REST services support the following authentication types:
  - HTTP Basic Authentication
  - Oracle E-Business Suite Security Token-Based Authentication

  For more REST service security, see "Managing Web Service Security" in Securing Web Services chapter of the *Oracle E-Business Suite Integrated SOA Gateway Implementation Guide*.

- REST services expect Oracle E-Business Suite security context values in the RESTHeader element of the HTTP requests.

- REST services support REST messages in XML or JSON format.

  Refer to "Understanding REST Messages" in Discovering and Viewing

Integration Interfaces chapter of the *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*.

Before using a deployed REST service for your custom app development, you need to test it first to ensure it can be invoked successfully once the required credentials and input payload are provided. See: Testing and Validating the REST Services, page 7-13.

Implementing Custom Interfaces

Creating Custom Interfaces

If there is no interface that meets your requirements or if you need additional interfaces for your custom mobile app, you can create custom interfaces of any interface types that can be exposed as REST services by using any technology available in Oracle E-Business Suite Integrated SOA Gateway. Annotate the custom interfaces based on the annotation standards and then upload them to the Integration Repository.

> **Note:** When you implement REST services for use in custom mobile apps, it is important to implement the interface in a way as lightweight as possible, that is, focusing on the specific data being retrieved or other operation being performed. It is not recommended that you wrap an existing interface for use in custom mobile apps if the exact internals of that interface are not known.

For more information about creating custom interfaces, refer to Administering Custom Integration Interfaces and Services chapter in the *Oracle E-Business Suite Integrated SOA Gateway Implementation Guide*. For annotation information and guidelines, see Appendix A Integration Repository Annotation Standards in the *Oracle E-Business Suite Integrated SOA Gateway Developer's Guide*. For information on uploading annotated custom interfaces to the Oracle Integration Repository, see "Generating and Uploading iLDT Files" and "Uploading ILDT Files to Integration Repository" in Administering Custom Integration Interfaces and Services chapter of the *Oracle E-Business Suite Integrated SOA Gateway Implementation Guide*.

Implementing Custom Interfaces as REST Services

Custom interfaces are categorized with Interface Source "Custom" in the Integration Repository, in contrast to Interface Source "Oracle" for Oracle seeded interfaces in Oracle E-Business Suite.

Once annotated custom interface definitions are successfully uploaded to the Integration Repository, you can locate them in the main Search page. First, click the **Show More Search Options** link, and then select "Custom" from the Interface Source drop-down list, along with desired business entity, interface type, or product family if needed before clicking **Go** to run the search.

After locating the desired custom interfaces from the repository either through a search or browse from the tree nodes, you can deploy the custom interfaces as REST services

by following the rest of the tasks as described earlier in Using Oracle Published Mobile APIs, page 7-4. These tasks include creating security grants, deploying the interfaces as REST services, and viewing the deployed WADL descriptions.

Testing and Validating the REST Services

Once a REST service is deployed on an Oracle E-Business Suite's application server, this deployed service is available and ready to be invoked.

Before you use the deployed REST service in a custom mobile app, it is important to test the REST service directly using a REST client, the same way it would be invoked in a mobile app flow. For example, a typical flow when a mobile user uses a mobile app is as follows:

1. Log in to the app.
2. Select a responsibility (and an organization in some apps) that initializes required application context.
3. Navigate mobile app pages that invoke REST services to retrieve and display data.
4. Log out of the app.

Following steps explain how the REST services can be tested:

1. Testing the Login Service, page 7-13
2. Testing the Initialize Service, page 7-14
3. Testing Your Mobile REST Services, page 7-17
4. Testing the Logout Service, page 7-18

If Oracle E-Business Suite Mobile Foundation Login component is used to develop custom mobile apps, you are not required to invoke the Login, Initialize and Logout services. These service invocations are pre-built into the seeded flow of the Login component where the user is required to log in to the app using the default login page, select a responsibility to initialize the Oracle E-Business Suite session context if required, and log out from Springboard. You are only required to design mobile app pages and invoke your app's REST services. For more details on how to develop custom apps using the Login component, refer to Using the Login Component to Develop Mobile Apps, page 5-1 and Using the Sample App as a Reference, page B-1.

If you do not use the Login component for your custom app development, then you will have to leverage the Login, Initialize, and Logout services as allowed by the development framework you plan to use.

Testing the Login Service

The Login service (`http://<hostname>:`

<port>/webservices/rest/AuthenticationService?WADL) authenticates Oracle E-Business Suite mobile app user's credentials based on the HTTP Basic authentication scheme. It takes the user name and password provided in the "Authorization" HTTP header of the GET request. The user name and password are concatenated as `username:password`, and encoded to base64 format. If the login is successful, it returns a unique access token (Oracle E-Business Suite session ID) as the `Set-Cookie` header and also in the body of HTTP response.

The access token from the Login service that points to the user session will be passed as the `Cookie` HTTP header in all subsequent REST service calls for user authentication, without the need for the user name and password to be sent every time.

To better understand how to test and validate the Login service, a sample request to the Login service and a response from the Login service are included as follows:

- A sample request to the Login service is as follows:

  ```
  GET /webservices/rest/login HTTP/1.1
  Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxx=
  Accept-Language: en-GB,en-US;q=0.8,en;q=0.6
  Content-Type: application/xml
  ```

  The HTTP headers used in this sample request are:

  - `Authorization`: Basic authentication scheme with base64 encoded `username:password`.
  - `Accept-Language`: To initialize the NLS language context with a given locale, send value in RFC 5646 language format. For example, `<lang>-<COUNTRY>`.
  - `Content-Type`: application/xml.

- A sample response from the Login service is as follows:

  ```
  Set-Cookie: demo=xxxxxxxxxxxxxxxxxxxxxxxxx
  Content-Type: application/xml

  <?xml version = '1.0' encoding = 'UTF-8' ?>
  <response>
   <data>
    <accessToken>xxxxxxxxxxxxxxxxxxxxxxxxx</accessToken>
    <accessTokenName>demo</accessTokenName>
    <ebsVersion>12.1.3</ebsVersion>
    <userName>[user_name]</userName>
   </data>
  </response>
  ```

  Note that a unique access token is available in the HTTP response as the `Set-Cookie` header and also in the `<accessToken>` element of the response body.

Testing the Initialize Service

Once the Login service succeeds, the Initialize service (`http://<hostname>:<port>/webservices/rest/AuthenticationService?WADL` with the POST HTTP method) is optionally invoked to initialize the Oracle E-Business Suite session

with required responsibility and organization contexts to authorize the application data access only to appropriate users.

For example, if an Oracle E-Business Suite mobile app requires the responsibility context in order to access specific Oracle E-Business Suite data, after the user login, a list of responsibilities is displayed. The user needs to select a responsibility (and an organization if needed) required for invoking the Initialize service.

Perform the following tasks to test and validate the Initialize service for the security context:

1. Identify the required security context for your app's REST services:
    - Responsibility context only

      If only the responsibility context is required, specify the Responsibility ID and Application ID to test the service.

      > **Note:** The Initialize service always initializes the responsibility context in the Oracle E-Business Suite session.

    - Responsibility context and organization context

      If both the responsibility context and organization context are required, specify the responsibility details with Organization ID.

      Apart from initializing the responsibility and organization contexts in the Oracle E-Business Suite session, some seeded REST services may require Organization ID to be passed as an input parameter to the REST services. In such cases, pass the selected Organization ID to those REST services as input so that the services can retrieve data using an appropriate organization context.

    - Any additional context other than the responsibility context or organization context that is required to be initialized for the app-specific REST services.

2. Invoke the Initialize service.
    - Following is a sample request to the Initialize service:

```
POST /webservices/rest/initialize HTTP/1.1
Cookie: demo=xxxxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/xml

<data>
 <resp>
  <id>20872</id>
  <applId>178</applId>
 </resp>
 <securityGroup>
  <id>0</id>
 </securityGroup>
 <org>
  <id>1733</id>
 </org>
</data>
```

The HTTP headers and input parameters used in this sample request are:

- `Cookie`: Access token from the Login service in the format of `<accessTokenName>=</accessTo>`
- Input payload
  - Responsibility details, such as Responsibility ID and Responsibility Application ID
  - Security group ID
  - Organization ID

  Note: Instead of passing values as ID, the key values can also be passed as the following example input payload.

  ```
  <data>
   <resp>
    <key>SYSTEM_ADMINISTRATION</key>
    <applKey>ICX</applKey>
   </resp>
   <securityGroup>
    <key>STANDARD</key>
   </securityGroup>
   <org>
    <key>CM1</key>
   </org>
  </data>
  ```

- Following is a sample response from the Initialize service.

```
Set-Cookie:
JSESSIONID=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/xml

<?xml version = '1.0' encoding = 'UTF-8' ?>
<response>
<data>completed with values resp_id=20872,
resp_key=SYSTEM_ADMINISTRATION, appl_id=178, appl_key=ICX,
sec_group_id=0, sec_group_key=STANDARD, org_id=1733,
org_key=Vision Communications (USA)</data>
</response>
```

Testing Your REST Services

Once your desired API is deployed as a REST service (`http://<hostname>:<port>/webservices/rest/<service_alias>?WADL`) from the Oracle Integration Repository, you can use the deployed REST service to fetch Oracle E-Business Suite data for your mobile app.

Perform the following tasks to invoke the REST service using the security token:

1. To invoke a REST service, insert the `Cookie` headers from the Login service. Create payload as required by the REST service.

   > **Note:** You can use any Integrated Development Environment (IDE), such as Oracle JDeveloper, to create a sample request payload from the XSD associated with the REST resource. References to different REST resources (APIs) and corresponding XSDs for request and response are available in the WADL file.

   A sample request payload for invoking a REST service is as follows:

```
POST /webservices/rest/<service_alias>/<rest_method> HTTP/1.1
Cookie: demo=xxxxxxxxxxxxxxxxxxxxxxxxxx
Accept-Language: en-GB,en-US;q=0.8,en;q=0.6
Content-Type: application/xml

<restMethod_Input xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
                  xmlns="http://xmlns.oracle.
com/apps/fnd/rest/servicealias/restMethod">
<InputParameters>
 <parameterOne>valueOne</parameterOne>
 <parameterTwo>valueOne</parameterTwo
</InputParameters>
</restMethod_Input>
```

   > **Note:** Once the Initialize service is invoked to initialize the required responsibility and organization contexts for a given session identified by a session token from the Login service, the contexts apply to all REST service invocations that use that session token. You must not include the `<RESTHeader>` element in the REST

service requests.

However, if your REST service requires application contexts that are different from the ones initialized using the Initialize service for the session, you must pass them as part of the `<RESTHeader>` element of that specific REST service request.

2. To perform the language testing for the REST service, use the Accept-Language HTTP header of the REST service request and send value in RFC 5646 language format. For example, `<lang>-<COUNTRY>`.

   Oracle E-Business Suite Integrated SOA Gateway always checks the language information of an incoming REST service call and resets the language in the Oracle E-Business Suite session. If no language is set in the REST header, then the session language defined through the ICX_LANGUAGE profile option is used as the Oracle E-Business Suite user's language.

Testing the Logout Service

The Logout service (`http://<hostname>:<port>/webservices/rest/AuthenticationService?WADL`) invalidates the current Oracle E-Business Suite session.

To invoke the Logout service, insert the `Cookie` headers from the Login service, especially **`Cookie:`**`<accessTokenName>=<accessToken>`.

A sample request for the Logout service is as follows:

```
GET /webservices/rest/logout HTTP/1.1
Cookie: demo=xxxxxxxxxxxxxxxxxxxxxxxxxx
```

To invalidate the current Oracle E-Business Suite session, send a GET request to the Logout service using the current session's security token in the `Cookie` header. The request does not require any payload.

Troubleshooting Tips

For troubleshooting information on potential problem symptoms and corresponding solutions, refer to the following documents for your Oracle E-Business Suite:

- For Oracle E-Business Suite Release 12.2, see Section 2.2: Troubleshooting Tips for REST Services in My Oracle Support Knowledge Document 1317697.1, *Oracle E-Business Suite Integrated SOA Gateway Troubleshooting Guide, Release 12.2*.

- For Oracle E-Business Suite Release 12.1.3, see My Oracle Support Knowledge Document 726414.1, *Oracle E-Business Suite Integrated SOA Gateway Troubleshooting Guide, Release 12*.

8

Internationalizing Oracle E-Business Suite Mobile Apps

Overview

This chapter provides required guidelines to internationalize Oracle E-Business Suite REST services and mobile apps. If your mobile users belong to different regions and prefer different languages, it is important to implement the mobile apps that can adapt to the user preferences.

Guiding Principles of Internationalizing Mobile Apps

When internationalizing mobile apps, you should consider the following:

- When developing mobile apps, you must implement required formatting in the code for internationalization. All Oracle E-Business Suite mobile apps have to work properly in various locales and character sets. For example,
  - A Date value should be formatted differently for different locales.
  - User data should be handled correctly regardless of its character set.
- User interface must be translated without having to change code.

Oracle E-Business Suite Mobile Foundation provides required infrastructure to implement internationalized mobile apps through its Login component. In some areas where an API is not available through the Login component, follow the guidelines described in this chapter to implement internationalization. For additional information, refer to Localizing MAF Applications, *Developing Mobile Applications with Oracle Mobile Application Framework*.

For information about the available languages in Oracle E-Business Suite mobile apps, see Setting Up and Using the Supported Languages, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

This chapter includes the following topics:

- Implementing REST Services, page 8-2
- Implementing Mobile Apps, page 8-7
- Known Issues and Limitations, page 8-17

Implementing REST Services

Data exchange between Oracle E-Business Suite mobile apps and the Oracle E-Business Suite server should be implemented using REST services. Before a mobile app begins to retrieve data from Oracle E-Business Suite or save data to Oracle E-Business Suite, the Login component initializes the required server sessions.

For information on the sequence of service invocation and how to test and validate the REST services, see Testing and Validating the REST Services, page 7-13.

Handling Data to and from Oracle E-Business Suite

Use the following guidelines to ensure that data exchange between Oracle E-Business Suite mobile apps and Oracle E-Business Suite are internationalized.

- Data retrieved from Oracle E-Business Suite should be based on the session language initialized using the mobile device locale. This guarantees that when the data is displayed on a mobile app, it is translated as per device language preference.
- Data encoding of the REST input and output messages should be UTF-8 to cover all Oracle E-Business Suite supported languages.
- The REST input and output messages should follow the XSD standard. Locale sensitive data such as Date, Date Time, and Number types should be represented in appropriate canonical formats, as defined by the XSD standard (http://www.w3.org/TR/xmlschema11-2/#built-in-primitive-datatypes).
    - For datetime (type="xs:dateTime"), the ISO 8601 (http://www.w3.org/TR/NOTE-datetime) canonical format has to be used. For example, 2014-01-31T17:43:15Z (YYYY-MM-DDThh:mm:ss.sZ).
    - For Date type values, the mobile app should send the canonical Date string in the REST request to Oracle E-Business Suite. Similarly, Oracle E-Business Suite will return the canonical Date string in the REST response to the mobile app.
- Locale sensitive data from Oracle E-Business should be formatted on the mobile app only based on the default format pattern and timezone set on the mobile device. Such data should not be formatted on the Oracle E-Business Suite server.
- For handling currencies, the currency code should be in ISO 4217 format, such as `USD` for US Dollar instead of a currency symbol $, because the same currency

symbol can be used for different currencies.

Additionally, the currency code, such as `USD`, can be retrieved from the server but the corresponding amount value in the Number format should be retrieved from the server in the canonical format and then formatted on the mobile app based on device locale.

- Oracle E-Business Suite REST service implementation should understand REST request values, such as Date, Date Time, and Number in the canonical format.

  Important: Oracle E-Business Suite REST services provided through Oracle E-Business Suite Integrated SOA Gateway support REST request headers, such as the `<NLSLanguage>` and `<Language>` parameters, and other security context headers, such as the `<Responsibility>` and `<RespApplication>` parameters.

  When Oracle E-Business Suite mobile apps invoke the REST services, do not pass values for the REST services. For mobile apps, it is only recommended that you use the Login component to initialize the server session with appropriate security context and language, but not through a specific REST request.

Handling Date Type Value in Application Module Services

If Application Module-based REST services are used in your mobile apps, by default the Date value in a REST output is formatted as YYYY-MM-DD (for example, 2005-02-03) in which case the time part is 00:00:00 or as YYYY-MM-DD hh:mm:ss.s (1999-06-30 10:29:34.0). This is not the canonical ISO 8601 format. In order for Oracle MAF View Layer to display the Date value based on the device locale, the REST output should have Date value formatted in the ISO 8601 format.

To work around this issue, overwrite the underlying View Object's getter method as shown in the following example:

```
// Create a subclass of oracle.jbo.domain.Date.

import org.w3c.dom.Document;
import org.w3c.dom.Node;

import oracle.jbo.domain.Date;

public class DateISO extends Date {

    public DateISO(Date d) {
        super(d);
    }

    public Node getXMLContentNode(Document xmlDoc)
    {
        return
xmlDoc.createTextNode(Util.formatISO8601(timestampValue()));
    }

    public int compareTo(Date date)
    {
        return super.compareTo(date);
    }
}

// Util. formatISO8601 method would be implemented in a utility class
like this.

private static final String ISO_CANONICAL_FORMAT_MASK = "yyyy-MM-
dd'T'HH:mm:ss.SSS'Z'";
private static final Locale ISO_CANONICAL_LOCALE = Locale.US;
private static final TimeZone UTC_TIMEZONE = TimeZone.getTimeZone
("UTC");

public static String  formatISO8601(Date date) {
 SimpleDateFormat sdf = new SimpleDateFormat(ISO_CANONICAL_FORMAT_MASK,
   ISO_CANONICAL_LOCALE);
 sdf.setTimeZone(UTC_TIMEZONE);
 return sdf.format(date);
}

// Overwrite a ViewObject's getter method like this.

From:

public Date getHiredate() {
   return (Date)getAttributeInternal(HIREDATE);
}

To:

public Date getHiredate() {
   if (getAttributeInternal(HIREDATE) != null) {
      return new DateISO((Date)getAttributeInternal(HIREDATE));
   } else {
      return getAttributeInternal(HIREDATE);
   }
}
```

A Sample REST Request Message

A REST request message to Oracle E-Business Suite for a PL/SQL REST service could be like:

```
<?xml version="1.0" encoding="UTF-8" ?>
<n0:notificationdetails_Input xmlns:n0="http://xmlns.oracle.
com/apps/fnd/rest/mobileplsqlsample/notificationdetails">
    <RESTHeader>
       <Responsibility />
       <RespApplication />
       <SecurityGroup />
       <NLSLanguage />
       <Org_Id />
    </RESTHeader>
    <n0:InputParameters>
       <n0:NOTIFICATIONID>xxxxxxx</n0:NOTIFICATIONID>
    </n0:InputParameters>
</n0:notificationdetails_Input>
```

A Sample REST Response Message

A REST response message from Oracle E-Business Suite for the same PL/SQL REST service could be like:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<OutputParameters xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.oracle.
com/apps/fnd/rest/mobileplsqlsample/notificationdetails/">
 <NOTFDETAILS>
  <HEADER>
   <HEADER_ITEM>
    <NAME>Id</NAME>
    <VALUE>xxxxxxx</VALUE>
    <TYPE>TEXT</TYPE>
   </HEADER_ITEM>
   <HEADER_ITEM>
    <NAME>To</NAME>
    <VALUE>LAST_NAME, FIRST_NAME</VALUE>
    <TYPE>TEXT</TYPE>
   </HEADER_ITEM>
   <HEADER_ITEM>
    <NAME>Sent</NAME>
    <VALUE>2015-08-10T07:08:28Z</VALUE>
    <TYPE>DATA</TYPE>
   </HEADER_ITEM>
   <HEADER_ITEM>
    <NAME>Due</NAME>
    <VALUE>2016-08-09T07:08:28Z</VALUE>
    <TYPE>DATE</TYPE>
   </HEADER_ITEM>
  </HEADER>
  <DETAIL>
   <DETAIL_ITEM>
    <NAME>Document ID</NAME>
    <VALUE>APPROVAL_NOTIFICATION</VALUE>
    <TYPE>TEXT</TYPE>
   </DETAIL_ITEM>
   <DETAIL_ITEM>
    <NAME>Transaction Id</NAME>
    <VALUE>xxxxxx</VALUE>
    <TYPE>NUMBER</TYPE>
   </DETAIL_ITEM>
   <DETAIL_ITEM>
    <NAME>Process Display Name</NAME>
    <VALUE>Update to Offer</VALUE>
    <TYPE>TEXT</TYPE>
   </DETAIL_ITEM>
   <DETAIL_ITEM>
    <NAME>APPROVED</NAME>
    <VALUE>Approve</VALUE>
    <TYPE>TEXT</TYPE>
   </DETAIL_ITEM>
   </DETAIL>
  <RESULTS>
   <RESULTS_ITEM>
    <NAME>APPROVED</NAME>
    <VALUE>Approve</VALUE>
    <TYPE>TEXT</TYPE>
   </RESULTS_ITEM>
   <RESULTS_ITEM>
    <NAME>REJECTED</NAME>
    <VALUE>Reject</VALUE>
    <TYPE>TEXT</TYPE>
   </RESULTS_ITEM>
  </RESULTS>
 </NOTFDETAILS>
</OutputParameters>
```

Implementing Mobile Apps

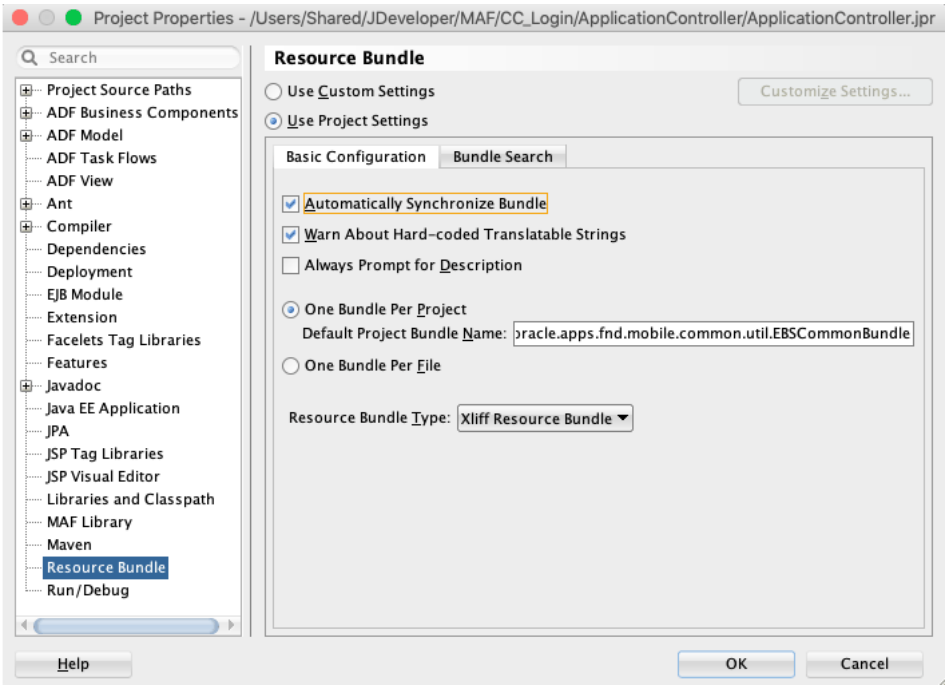
This section includes the following topics:

Configuring MAF Applications for Internationalization

After creating an Oracle Mobile Application Framework project, configure the MAF application for internationalization:

1. In Oracle JDeveloper, right click the ApplicationController project, and then **Project Properties**.
    - Select **Compiler** and select "UTF-8" in the Character Encoding field from the drop-down list.
    - Select **Resource Bundle** and specify the following information:
        - Select the "Automatically Synchronize Bundle" check box to enable the feature.
        - Select the "Warn About Hard-coded Translatable Strings" check box to enable the feature.
        - Do not select the "Always Prompt for Description" check box.
        - Leave the default selection in the "One Bundle Per Project" button and the default project bundle name unchanged.
        - Leave the default value "Xliff Resource Bundle" unchanged in the Resource Bundle Type field.

The Project Properties Window



2. Right click the ViewController project, and then **Project Properties**.

   Repeat the same tasks as described in step 1 to set the complier and resource bundle for the ViewController project.

For information on setting up an environment for internationalization, see Setting Up Oracle JDeveloper for Internationalization, page 3-26.

Translating Mobile App User Interface

If an Oracle E-Business Suite mobile app is designed to be used in different languages, it is important to ensure that the content is translated to required languages without having to make any changes to the code. In order to translate the mobile app, follow the general translation principles supported by Oracle.

General Translation Principles

- All information displayed to mobile app users, such as names, labels, and values must be translated. It includes:
  - UI component labels or values used in the AMX pages
  - Feature names in the `maf-feature.xml` file

    - Application name in the `maf-application.xml` file
    - Application preference labels in the `maf-application.xml` file

- Use XLIFF (XML Localization Interchange File Format) resource bundle to hold translatable strings and reference them in the User Interface elements as necessary.
- The XLIFF file should be encoded in UTF-8.
- All values retrieved from the Oracle E-Business Suite server through REST services to shown to users should be retrieved in the current session language.
- All values coded in the mobile app that are shown to users should come from XLIFF resource bundle.

```
<!--AVOID THIS-->
<amx:outputText value="Enter Username" id="ot6"/>

<!--RECOMMENDED-->
<amx:loadBundle basename="mobile.testPageBundle" var="
viewcontrollerBundle" id="lb1" />
...
<amx:outputText value="#{viewcontrollerBundle.ENTER_USERNAME}" id="ot1"
/>
```

Designing for Translation

In order for the mobile apps to be translated correctly, consider the following guidelines:

- Translation expansion.

  Some languages require more space on average, compared with English text. Strings that are too long or screens that are already crowded in English may cause serious usability issues after translation, such as truncations and other layout issues. To avoid such issues, consider minimum space available for translation to be the length of the English text + 30% + 2 characters.

- Text wrapping due to translation expansion should be weighed against a potential negative impact on user experience, by increasing the need for scrolling.
- Characters (font) in several Asian languages tend to be taller than those in Latin-based languages.
- Due to the limited display space in a mobile device, whole text may not be displayed. In this situation, one possible solution is to truncate a user interface label or text with ellipsis like "Label is truncat...".

  This ellipsis highlights the cases where text overflows the available or designated space. You can rotate your mobile device to landscape mode to see the full text.

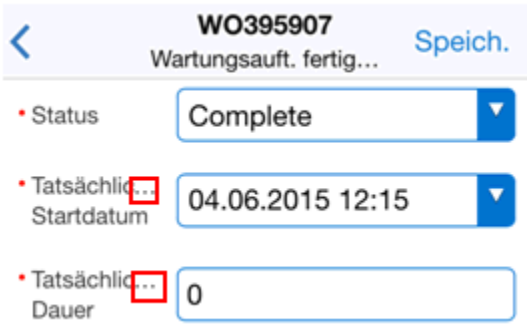
- Avoid concatenating translatable messages, as this will adversely affect translation quality.

```
<!--AVOID THIS-->
<amx:outputText value="#{bean.diskName} #{bundle.CONTAINS} #{bean.
fileNumber} #{bundle.FILES}" id="it6"/>
```

Handling Text Overlap and Truncation

Due to the limited display space in a mobile device, whole text may not be displayed. In this situation, one possible solution is to truncate a user interface label or text with ellipsis.

Examples of Truncated Label with Ellipsis



This ellipsis highlights the cases where text overflows the available or designated space. You can rotate your mobile device to landscape mode to see the full text.

Mobile Device in Landscape Mode with Full Text

Check the following style attributes to the component you want to have this truncation with ellipsis handling:

- text-overflow: ellipsis;
- overflow: hidden;
- display: block; or (display: inline-block;)

- white-space: nowrap;
- max-width: 100% or (some other value). This attribute is needed some cases.

For example, like the field label case described earlier, you can get the field label truncation with ellipsis by setting the `text-overflow: ellipsis;` attribute to the `.amx-listItem .amx-outputText` style.

```
.amx-listItem .amx-outputText {
  text-overflow: ellipsis;
}
```

Translating XLIFF files

1. Use the base XLIFF file which contains the source strings to generate the translated XLIFF files.
2. The translated XLIFF file name should adhere to the following naming conventions:

   `<BASE_XLIFF_FILE_NAME>_<LANGUAGE_TOKEN>.xlf`
   Where:

   - `<BASE_XLIFF_FILE_NAME>` is the base XLIFF file name, without the .xlf extension.
   - `<LANGUAGE_TOKEN>` is the lower case ISO 639 two-letter language code and ISO 3166 two-letter country code if needed to identify the language, such as zh_TW (Traditional Chinese).
3. For example, for a base XLIFF named `viewcontrollerBundle.xlf`, the translated XLIFF file names for English (Base), Korean, and Traditional Chinese languages are respectively:
   - English (Base xliff file, which is not translated): `viewcontrollerBundle.xlf`
   - Korean: `viewcontrollerBundle_ko.xlf`
   - Traditional Chinese: `viewcontrollerBundle_zh_TW.xlf`
4. Place the translated XLIFF files in the same directory as the corresponding base XLIFF file.

Implementing Model Layer

Mapping REST Output

The REST output from Oracle E-Business Suite should be mapped to a Java object (such as canonical Date type value maps to a Java Date object, canonical Number type value maps to a Java Number object, etc.) based on the XSD associated with the REST output.

No special handling is needed to get a Java object from the REST data.

Avoid formatting Date, Date Time, and Number types in the model layer using Java formatting, such as the following.

- java.text.DateFormat
- java.text.SimpleDateFormat
- java.text.NumberFormat
- java.text.DecimalFormat

Formatting Date, Date Time, Number should be done in the view layer (AMX layer).

Accessing XLIFF Resource Bundle

Use `oracle.adfmf.util.BundleFactory.getBundle` to get the `java.util.ResourceBundle` object, then call the `ResourceBundle.getString` method.

```
ResourceBundle rb = BundleFactory.getBundle("PATH.TO.RESOURCE.BUNDLE");
String errorMsg = rb.getString("ERROR_MESSAGE");
```

Implementing View Layer

Use AMX and/or AMXF

- Use AMX and/or AMXF (AMX Page Fragment) to display content of a feature.
- Avoid local HTML and remote URL, which could show locale sensitive data (Date, Date Time, Number) differently from the AMX/AMXF based pages.

Layout, Font, Text, and Style

- Alignment should not be included in the code such as the `inlineStyle` attribute or style sheet (CSS file) unless required in a special case to support BiDi languages. Avoid using the following style properties to have the horizontal alignment.
  - text-align
  - padding-left
  - padding-right
  - padding
  - margin-left
  - margin-right
  - margin

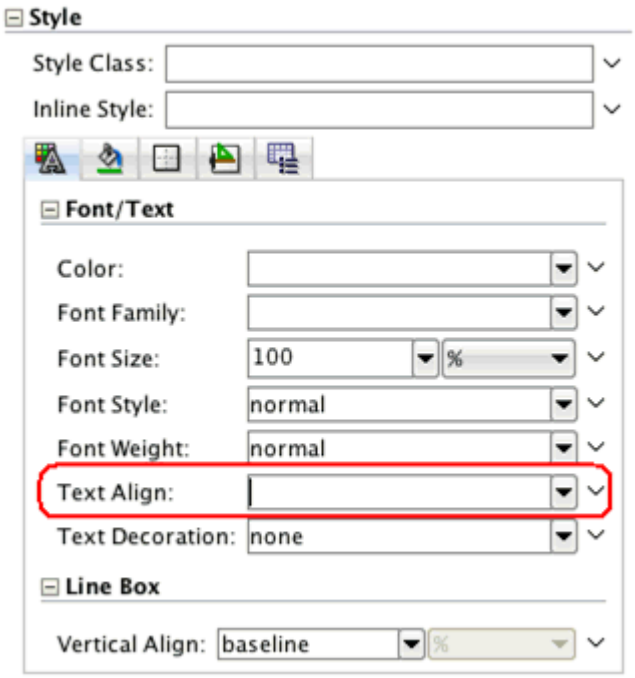
- float
- position
- background-position
- border-left
- border-left-color
- border-left-style
- border-left-width
- border-right
- border-right-color
- border-left-style
- border-left-width

For example, avoid including "left" or "right" in the code for "text-align" in the `inlineStyle` attribute.

```
<!--AVOID THIS -->
<amx:outputText value="#{viewcontrollerBundle.HIRE_DATE}"
          inlineStyle="text-align:left;" id="it6"/>
```

- Leave it blank in the "Text Align" property in the Style section of the Property Inspector panel.

Style Section with "Text Align" Property Highlighted

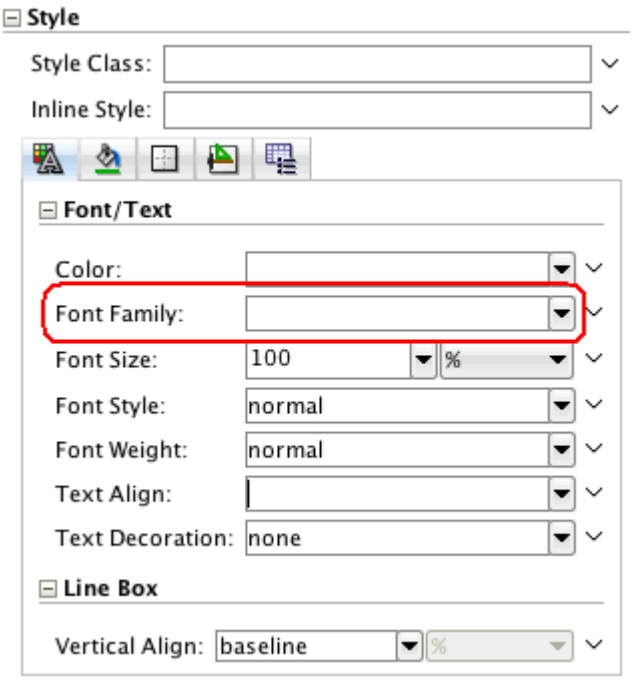


Avoid specifying "font-family" in the `inlineStyle` attribute, as shown in the following example of incorrect usage.

```
<!--AVOID THIS -->
<amx:outputText value="#{viewcontrollerBundle.HIRE_DATE}"
          inlineStyle="font-family:Tahoma;" id="it6"/>
```

Instead, leave it blank in the "Font Family" property of the Style section in the Property Inspector panel.

Style Section with "Font Family" Property Highlighted



Time-based Conversion and Formatting

In general, let MAF format a date, date time, or time with the default pattern. The default pattern changes depending on the mobile device locale setting. Avoid specifying a pattern as it may not work for certain languages or regions.

```
<!--RECOMMENDED -->
<amx:outputText value="#{bindings.hireDate.inputValue}"
           label="#{viewcontrollerBundle.HIRE_DATE}" id="it6">
<amx:convertDateTime pattern=""/>
<!-- No pattern was specified; default pattern is being used -->
</amx:outputText>

<!--AVOID THIS -->
<amx:outputText value="#{bindings.hireDate.inputValue}"
           label="#{viewcontrollerBundle.HIRE_DATE}" id="it6">
   <amx:convertDateTime pattern="yyyy-MMM-dd"/>
   <!-- DO NOT hard code pattern attribute -->
</amx:outputText>

<!--AVOID THIS -->
<amx:outputText value="#{bindings.hireDate.inputValue}"
           label="#{viewcontrollerBundle.HIRE_DATE}" id="it6">
   <amx:convertDateTime pattern="#{bindings.lastUpdateDate.format}"/>
   <!-- DO NOT pass a value to pattern attribute -->
</amx:outputText>
```

If a fixed pattern is needed to meet specific business requirements, specify the required pattern as follows:

```
<!--OK -->
<amx:outputText value="#{bindings.hireDate.inputValue}"
          label="#{viewcontrollerBundle.HIRE_DATE}" id="it6">
<amx:convertDateTime pattern="yyyy"/>
</amx:outputText>
```

- <amx:convertDateTime> has the datestyle and timestyle attributes whose values are "full", "long", "medium", "short", etc. If no style is specified, the "short" style is used as a default. Set the proper style based on your business requirements.

```
<!--OK -->
<amx:outputText value="#{bindings.hireDate.inputValue}"
           label="#{viewcontrollerBundle.HIRE_DATE}" id="it6">
<amx:convertDateTime pattern="" datestyle="long" timestyle="long" />
</amx:outputText>
```

Time-based Input and Date Picker

- Use <amx:inputDate> for date, datetime, and time data input with the date and time picker. Do not use <amx:inputText> because this would need to be used in conjunction with <amx:convertDateTime>, which is only supported for use with output.

```
<!--RECOMMENDED -->
<amx:inputDate value="#{bindings.hireDate.inputValue}"
               label="#{viewcontrollerBundle.HIRE_DATE}" id="it6"/>

<!--AVOID THIS -->
<amx:inputText value="#{bindings.hireDate.inputValue}"
               label="#{viewcontrollerBundle.HIRE_DATE}" id="it6">
   <amx:convertDateTime pattern=""/>
</amx:outputText>
```

- Use <amx:inputDate> to format or parse date, datetime, or time, based on the device locale settings. <amx:inputDate> does not have the pattern attribute, and cannot include <amx:convertDateTime>.

Calendar

- <amx:convertDateTime> always uses the Gregorian calendar, regardless of the mobile device's calendar setting. In contrast, <amx:inputDate> honors the mobile device's calendar setting.

- The Gregorian calendar should be set for mobile client devices to avoid showing different calendar dates in the UI pages.

Timezone

- The mobile device's timezone setting should be used.

- Use <amx:outputText> and <amx:convertDateTime> for date, datetime, and time conversion and formatting. Use <amx:inputDate> for date, datetime, and time data input. Depending on the mobile device's timezone setting, a date object is automatically converted to a formatted string.

Hour or 24 Hour Time Format

The mobile device's setting should be honored. Let AMF format the datetime or time so that the mobile device's setting is used.

Number Conversion and Formatting

- Use `<amx:convertNumber>` to format a number. Allow MAF to format the number with the default numeric characters (group and decimal separators). The default numeric characters would change based on the mobile device locale setting.

- Numbers
  - Use `<amx:outputText>` or `<amx:inputText>` and set the "type" attribute to "number" (or do not specify type attribute value). Depending on your specific business requirements, the following attributes of `<amx:convertNumber>` can be set for the number display except for currency number.
    - Grouping Used
    - Integer Only
    - Min Integer Digits
    - Max Integer Digits
    - Min Fraction Digits
    - Max Fraction Digits
  - `<amx:inputNumberSlider>` can be used for number input.

Known Issues and Limitations

- Only the Gregorian calendar is supported in the device calendar setting. If you use a non-Gregorian calendar such as Japanese or Buddhist on an iOS device, you most likely will see a mixture of calendars in mobile app pages.

- BiDi languages are not supported. Setting a BiDi language, such as Arabic or Hebrew, in the device setting is not supported.

- On iOS, you need to set the same language for the iOS language and preferred language. Using different languages for the iOS language and preferred language could result in mixture language UI where UI labels are shown in the iOS language, whereas language data from Oracle E-Business Suite is shown in the preferred language.

- The scope of Oracle E-Business Suite language support and the mobile device language support may be different. For example, iOS does not support Albanian

and Slovenian, but Oracle E-Business Suite does.

- If you have a MLS-enabled (multiple language support) Oracle E-Business Suite environment and you set your mobile device language to a language supported by Oracle E-Business Suite, but not in the languages listed as the supported languages for Oracle E-Business Suite mobile apps, then the data retrieved from the Oracle E-Business Suite server will be displayed in the mobile device specified language. However, the user interface labels within the app will appear in English.

- If you set your mobile device language to a language not supported by Oracle E-Business Suite or not enabled in your Oracle E-Business Suite environment, then the data coming from the Oracle E-Business Suite server will be displayed in the Oracle E-Business Suite base language.

- Supplementary characters are not supported because this is not supported by Oracle E-Business Suite.

9

Implementing Corporate Branding and Deploying Your Apps

Overview

To implement corporate branding to replace the Oracle logos with your enterprise logo, you must change:

- App logo
- Company logo
- Splash screen

Additionally, you can change the following attributes for your mobile apps:

- App Name
- End User License Agreement
- Copyright

The following table lists the corporate branding options available for Oracle E-Business Suite mobile apps:

Allowed Artifact Modification	Impacted Mobile App Page or Icon	Associated Change Instruction

App Logo	• Updated through a deployment profile:
	  • Marketing icon for an iOS app
	  • Application icon shown on a mobile device's home screen for an app, including both iOS and Android
	  • Notifications icon for an app on an iOS device
	  • Settings icon against an app name on an iOS device
	  • Spotlight search on an iOS device
	• Updated in the Oracle JDeveloper project:
	  • About page of the app for both iOS and Android
	  • Page header on an Android device
		• Customizing App Logo through a deployment profile:
		  1. Creating a New Deployment Profile, page 9-11
		  2. Changing the App Logo and Splash Screen, page 9-19
		• Customizing App Logo through an Oracle JDeveloper project:
		  Changing App Logo in the About Page and Android Page Header, page 9-25

Splash Screen	• Shown after the Sign In screen • Updated in maf-application. xml for iOS • Updated through a deployment profile for Android	1. Creating a New Deployment Profile, page 9-11 2. Changing the App Logo and Splash Screen, page 9-19

Allowed Artifact Modification	Impacted Mobile App Page or Icon	Associated Change Instruction
Company Logo	• Sign In screen • Server URL screen	Implementing Your Company Logo, page 9-27
App Name (Optional)	• Sign In screen • Server URL screen • Springboard • About page	Changing the App Name (Optional), page 9-30
Legal Terms including End User License Agreement and Copyright (Optional)	• End User License Agreement shown during the initial launch of an app • Copyright shown in the About page	• Customizing End User License Agreement or Legal Terms (Optional), page 9-35 • Customizing the Copyright in the About Page (Optional), page 9-37

The image sizing requirements for app logo, company logo, and splash screen listed in the table are described in the next section, Required Image Sizing Information for Corporate Branding, page 9-4.

For step-by-step instructions to implement corporate branding, see:

- Creating a New Deployment Profile, page 9-11
- Changing the App Logo and Splash Screen, page 9-19
- Changing the App Logo in the About Page and Android Page Header, page 9-25
- Implementing the Company Logo, page 9-27
- Changing the App Name (Optional), page 9-30
- Customizing the Legal Related Information (Optional), page 9-35

After implementing corporate branding for your app, you can deploy the app. See: Deploying Your Apps, page 9-39.

Required Image Sizing Information for Corporate Branding

Before you begin the implementation for corporate branding, prepare the images or icons based on the sizing requirements described in the following sections:

- Required Image Sizing Information for the iOS Platform, page 9-4
- Required Image Sizing Information for the Android Platform, page 9-7
- Common Sizing Requirements for the iOS and Android Platforms, page 9-10

Please note that all dimensions for images described in this section are in pixels.

Required Image Sizing Information for the iOS Platform

This section describes custom icons in specific sizes for application images if required to implement corporate branding for the iOS platform.

- Marketing Icon
- Application Icons
- Notifications
- Spotlight
- Settings

For more information on the iOS application icons images, see the Icon and Image Design section in the iOS Human Interface Guidelines from Apple iOS Developer Portal.

> **Note:** Oracle E-Business Suite mobile apps use squared corner icons for the iOS platform since iOS automatically rounds the icon corners.

For required image sizing information for app logo and company logo, see Common Sizing Requirements for the iOS and Android Platforms, page 9-10.

Sizing Requirements for Marketing Icon

Marketing icon image must be in `.png` format and 1024 x 1024 pixels.

Sizing Requirements for Application Icons

The following table describes the sizing information for application icons:

> **Note:** To easily distinguish the custom icons for corporate branding from the default icons used in the Oracle E-Business Suite mobile apps,

a prefix "XXX_" is added to the custom icon file names.

Sizing Requirements for Application Icons

Device Type	Description	Size	Recommended File Name
iPhone App 2x	Application icon for iPhone/iPod touch Retina	120x120	XXX_Icon-120.png
iPhone App 3x	Application icon for iPhone 6 Plus	180x180	XXX_Icon-60@3x.png
iPad App 1x	Application icon for iPad Non-Retina	76x76	XXX_Icon-76.png
iPad App 2x	Application icon for iPad Retina	152x152	XXX_Icon-76@2x.png
iPad Pro App 2x	Application Icon for iPad Pro Retina	167x167	XXX_Icon-83.5@2x.png

Sizing Requirements for Notifications

The following table describes the sizing information for notifications:

Sizing Requirements for Notifications

Device Type	Description	Size	Recommended File Name
iPad Notifications 1x	Notifications icon for iPad Non-Retina	20x20	XXX_Icon-20.png
iPad Notifications 2x	Notifications icon for Retina 2x devices	40x40	XXX_Icon-40.png
iPhone Notifications 2x	Notifications icon for Retina 2x devices	40x40	XXX_Icon-40.png

Device Type	Description	Size	Recommended File Name
iPhone Notifications 3x	Notifications icon for Retina 3x devices	60x60	XXX_Icon-60.png

Sizing Requirements for Spotlight

The following table describes the sizing information for spotlight:

Please note all the spotlight icons should be squared corner.

Sizing Requirements for Spotlight

Device Type	Description	Size	Recommended File Name
iPad Spotlight 1x	Spotlight icon for iPad Non-Retina	40x40	XXX_Icon-40.png
iPad Spotlight 2x	Spotlight icon for Retina 2x devices	80x80	XXX_Icon-40@2x.png
iPhone Spotlight 2x	Spotlight icon for Retina 2x devices	80x80	XXX_Icon-40@2x.png
iPhone Spotlight 3x	Spotlight icon for Retina 3x devices	120x120	XXX_Icon-40@3x.png

Sizing Requirements for Settings

The following table describes the sizing information for settings:

Please note all the settings icons should be squared corner.

Sizing Requirements for Settings

Device Type	Description	Size	Recommended File Name
iPad Settings 1x	Settings icon for iPad Non-Retina	29x29	XXX_Icon-29.png

Device Type	Description	Size	Recommended File Name
iPad Settings 2x	Settings icon for Retina 2x devices	58x58	XXX_Icon-29@2x.png
iPhone Settings 2x	Settings icon for Retina 2x devices	58x58	XXX_Icon-29@2x.png
iPhone Settings 3x	Settings icon for Retina 3x devices	87x87	XXX_Icon-29@3x.png

For instructions on customizing the app icons, splash screen, spotlight, and settings, see:

- Instructions to Create a New Deployment Profile for the iOS Platform, page 9-11
- Changing App Logo and Splash Screen, page 9-19

Required Image Sizing Information for the Android Platform

This section describes custom icons in specific sizes for application images if required to implement corporate branding for the Android platform.

> **Note:** Oracle E-Business Suite mobile apps use rounded corner icons for Android apps since Android does not round them automatically.

- Application Icon
- Splash Screen (Portrait)
- Splash Screen (Landscape)

For the required image sizing information for the app logo and company logo, see Common Sizing Requirements for the iOS and Android Platforms, page 9-10.

Sizing Requirements for the Application Icon, Splash Screen (Portrait), and Splash Screen (Landscape)

The following table describes the image sizing information for the application icon, splash screen (portrait), and splash screen (landscape):

> **Note:** To easily distinguish the custom icons for corporate branding from the default icons used in the Oracle E-Business Suite mobile apps, a prefix "XXX_" is added to the custom icon file names as shown in this table.

Sizing Requirements for the Application Icon, Splash Screen (Portrait), and Splash Screen (Landscape)

Platform Display Type	Icon	Size	Recommended File Name
Low Density	Application Launch Icon	36x36	XXX_display-ldpi-icon.png
	Splash Screen (Portrait)	480x800	XXX_display-port-ldpi-splashscreen.png
	Splash Screen (Landscape)	800x480	XXX_display-land-ldpi-splashscreen.png
Medium Density	Application Launch Icon	48x48	XXX_display-mdpi-icon.png
	Splash Screen (Portrait)	480x800	XXX_display-port-mdpi-splashscreen.png
	Splash Screen (Landscape)	800x480	XXX_display-land-mdpi-splashscreen.png
High Density	Application Icon	72x72	XXX_display-hdpi-icon.png
	Splash Screen (Portrait)	800x1280	XXX_display-port-hdpi-splashscreen.png
	Splash Screen (Landscape)	1280x800	XXX_display-land-hdpi-splashscreen.png
Extra High Density	Application Icon	96x96	XXX_display-xhdpi-icon.png
	Splash Screen (Portrait)	768x1024	XXX_display-port-xhdpi-splashscreen.png

Platform Display Type	Icon	Size	Recommended File Name
	Splash Screen (Landscape)	1024x768	XXX_display-land-xhdpi-splashscreen.png
Extra Extra High Density (available from Oracle E-Business Suite Mobile Foundation Release 5.0 only)	Application Icon	144x144	XXX_display-xxhdpi-icon.png
	Splash Screen (Portrait)	960x1600	XXX_display-port-xxhdpi-splashscreen.png
	Splash Screen (Landscape)	1600x960	XXX_display-land-xxhdpi-splashscreen.png
Extra Extra Extra High Density (available from Oracle E-Business Suite Mobile Foundation Release 5.0 only)	Application Icon	192x192	XXX_display-xxxhdpi-icon.png
	Splash Screen (Portrait)	N/A	N/A
	Splash Screen (Landscape)	N/A	N/A

For instructions on customizing the application logo and splash screen, see:

- Instructions to Create a New Deployment Profile for the Android Platform, page 9-16
- Changing App Logo and Splash Screen, page 9-19

Common Sizing Requirements for the iOS and Android Platforms

This section includes the following sizing requirements for the iOS and Android platforms:

- App Logo
- Company Logo

Sizing Requirements for the App Logo

The following table describes the sizing information for the app logo:

- Similar to the application's launch icon, the app logo uses the same rounded corners.
- Image files should be in `.png` format.

Sizing Requirements for the App Logo

Use	Size	Recommended File Name
App Logo on About page	152x152	Add a prefix "XXX_" to the custom app logo file name
App Logo on Page Header (Android Only)	80x80	android_app_header_icon.png

For instructions on customizing the application logo in the About page and page header, see Changing the App Logo in the About Page and Android Page Header, page 9-25.

Sizing Requirements for the Company Logo

The following table describes the sizing information for the company logo:

Sizing Requirements for the Company Logo

Description	Size
Maximum width	400
Minimum width	100

Description	Size
Maximum height	200
Minimum height	50

Once the company logo is created, name the image file as `CorporateLogo.png`.

For instructions on how to change the company logo, see Customizing the Company Logo, page 9-27.

Creating a New Deployment Profile

You can implement corporate branding for your mobile app with custom app logos or images through a new deployment profile.

This section provides information on how to create a new deployment profile for each platform.

- Instructions to Create a New Deployment Profile for the iOS Platform, page 9-11
- Instructions to Create a New Deployment Profile for the Android Platform, page 9-16

Instructions to Create a New Deployment Profile for the iOS Platform

To implement corporate branding for your app, you need to create a new deployment profile first, and then replace the default Oracle image used for the app icons with custom images. All images should be in `.png` format.

Copying the Custom Icons

Before creating a new deployment profile, copy the custom icon files to the following location where the default icons are also placed:

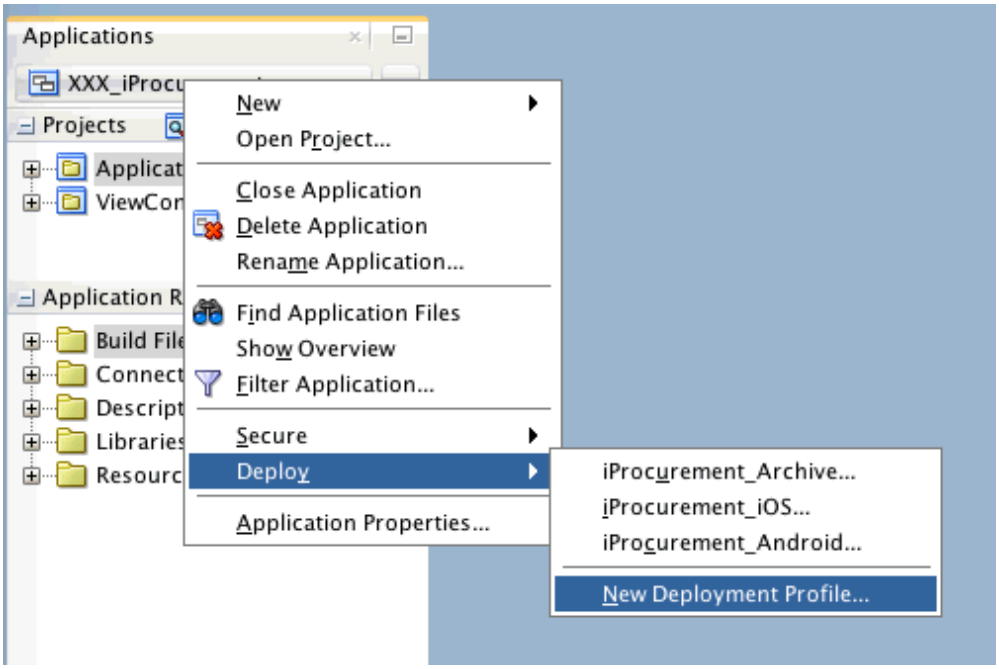
```
<Application Root Folder>/resources/ios
```

Creating a New Deployment Profile

Perform the following steps to create a new deployment profile:

1. In the Applications Navigator of Oracle JDeveloper, select "XXX_iProcurement" application and then right-click on it. Select **Deploy** and then **New Deployment Profile...** from the selection window.

Oracle JDeveloper Navigation Menu with "New Deployment Profile" Selected



2. In the Create Deployment Profile page, enter the following information:

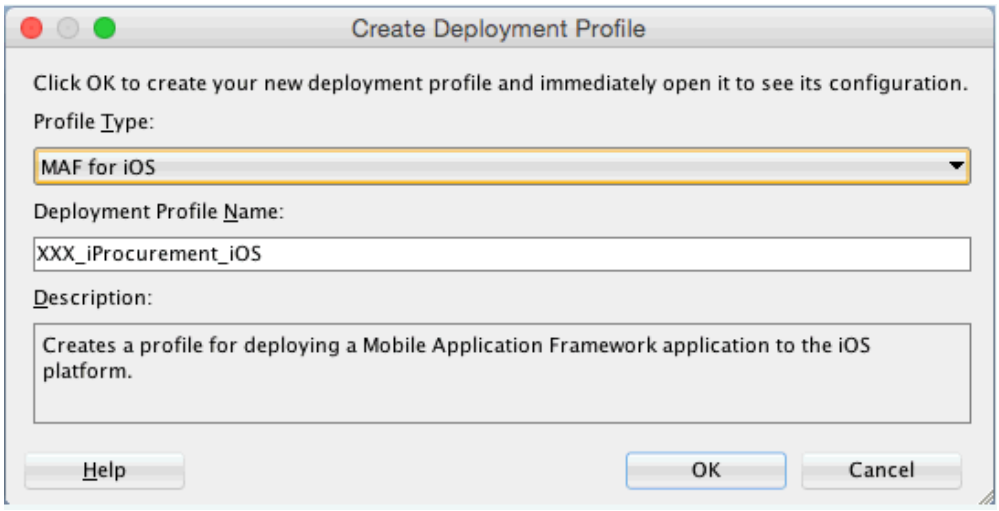
   - Select **MAF for iOS** as the Profile Type field.

     If "MAF for Android" is selected as the deployment profile type, see Instructions to Create a New Deployment Profile with Customized App Logos for the Android Platform, page 9-16.

   - Enter a new name, such as "XXX_iProcurement_iOS", as the Deployment Profile Name if required.

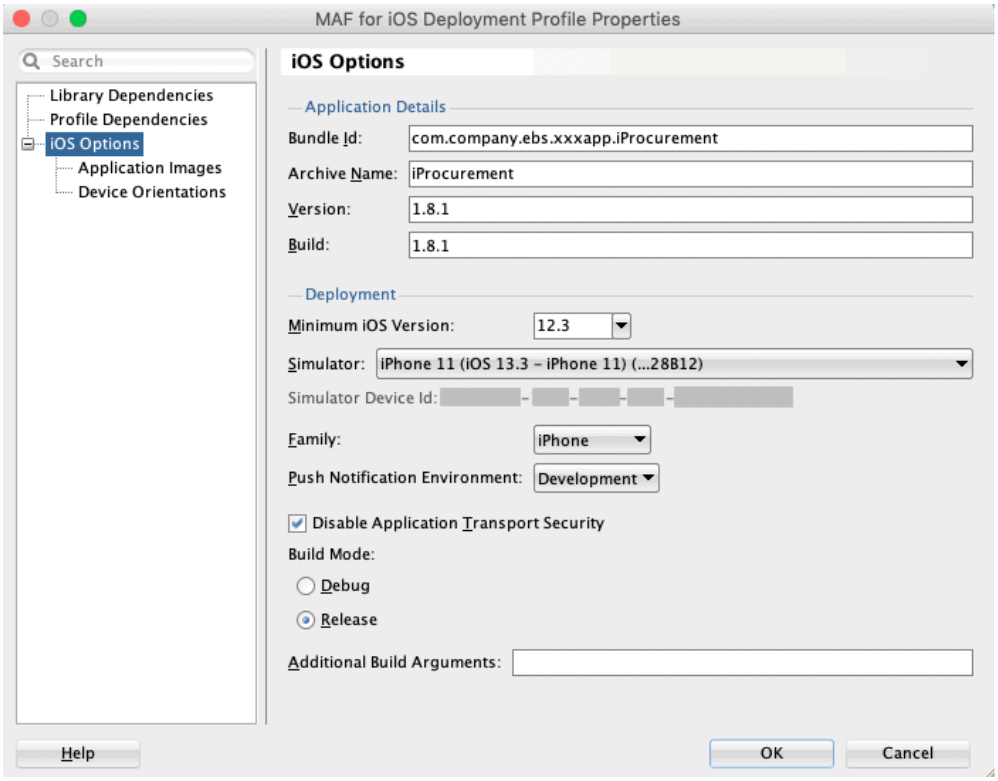
   Click **OK**.

Create Deployment Profile Page for iOS



3. Enter the following information in the MAF for iOS Deployment Profile Properties page:

MAF for iOS Deployment Profile Properties Page



- Bundle Id: Use this application Bundle Id to package the application binary for iOS, such as `com.company.ebs.xxxapp.iProcurement`.

  Please note that the Id must be a unique Id for each app installed on an iOS mobile device and must follow the reverse-package style naming conventions:

  `com.<companyname>.<organizationname>.<appname>`

  - If you deploy the app to an iOS device, the application Bundle Id should match the iOS provisioning profile that you receive after registering with the Apple's iOS Developer Program.

  - If you deploy the app to an iOS simulator for testing the mobile app, you can use any application Bundle Id of your choice but you could use the same application Bundle Id required by the iOS provisioning profile.

  The Bundle Id is used to package the application binary for iOS. Please note that the Bundle Id must be unique for each app installed on an iOS or Android device. Even if the same app has been deployed twice for an enterprise, a different Bundle Id uniquely represents each individual app. For example, one app can be used for production and the other can be used for testing purposes if desired.

Additionally, the application Bundle Id used here impacts only the application binary packaging and its installation on mobile devices. It does not have any impact on the Oracle E-Business Suite server. For example, you could have packaged two apps with two different application Bundle Ids in the MAF deployment profiles, such as `com.company.ebs.xxxapp.XXiProcurement` and `com.company.ebs.xxxapp.YYiProcurement`, but both could have the same Id in the **`maf-application.xml`** file, such as `com.company.ebs.xxxapp.iProcurement`. Both apps can be installed on the same mobile device and can connect to the same Oracle E-Business Suite server using one registration on the Oracle E-Business Suite server.

This is useful that you create the app registration once on the server, but you can download and apply it to other Oracle E-Business Suite instances. You can then use multiple installations of the same app on a single test device and test against different Oracle E-Business Suite instances.

- Archive Name: Enter the name of the .ipa file created by MAF. For example, iProcurement.

- Minimum iOS Version: This value should be 12.3.

- Simulator: Select from a list of iOS simulators to which you want to deploy the mobile app.

- Family: iPhone. Oracle E-Business Suite mobile apps are designed and tested for iPhones.

  Note that mobile users can run the mobile apps on any devices that are capable of running iOS 12.3 or later. However, the UI design is specific to iPhones only.

- Disable Application Transport Security: If your Oracle E-Business Suite Web Entry Point is TLSv1.2-enabled and it meets the certificate and cipher suite in Apple's Requirements for Connecting Using ATS (App Transport Security) (https://developer.apple.com/library/content/documentation/General/Reference/InfoPlistKeyReference/Articles/CocoaKeys.html#//apple_ref/doc/uid/TP40009251-SW35), then leave this box unchecked. Otherwise, select this check box.

  > **Note:** For information about whether Oracle E-Business Suite Release 12.1 and Release 12.2 using Oracle HTTP Server (OHS) as a termination point meets the forward secrecy certificate and cipher suite requirements for ATS, review the following TLS documents for updates in this area:
  >
  > - My Oracle Support Knowledge Document 376700.1, *Enabling TLS in Oracle E-Business Suite Release 12.1*

- My Oracle Support Knowledge Document 1367293.1, *Enabling TLS in Oracle E-Business Suite Release 12.2*

This check box is available for mobile apps built with Oracle E-Business Suite Mobile Foundation 5.0 and onwards.

- Build Mode: Select the **Debug** radio button for development and testing. Select the **Release** radio button for production deployment.

Click **OK**.

4. Save your work.

Instructions to Create a New Deployment Profile for the Android Platform

Similar to the creation of a new iOS deployment profile, when implementing corporate branding for an app, you need to create a new deployment profile for the Android platform first, and then replace the default Oracle image used for the app icons with custom images. All images should be in `.png` format.

Copying the Custom Icons

Before creating a new deployment profile, copy the custom icon files to the following location where the default icons are also placed:

```
<Application Root Folder>/resources/android
```

Creating a New Deployment Profile

Use the following steps to create a new deployment profile:

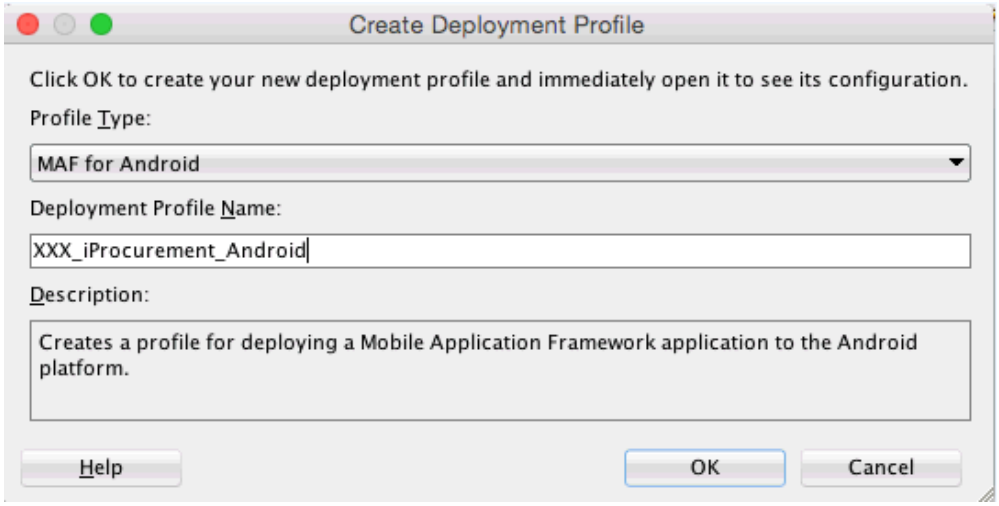
1. In the Applications Navigator of Oracle JDeveloper, select "XXX_iProcurement" application and then right-click on it. Select **Deploy** and then **New Deployment Profile...** from the selection window.

2. In the Create Deployment Profile page, select **MAF for Android** as the Profile Type field.

   If "MAF for iOS" is selected as the deployment profile type, see Instructions to Create a New Deployment Profile with Customized App Logos for the iOS Platform, page 9-11.

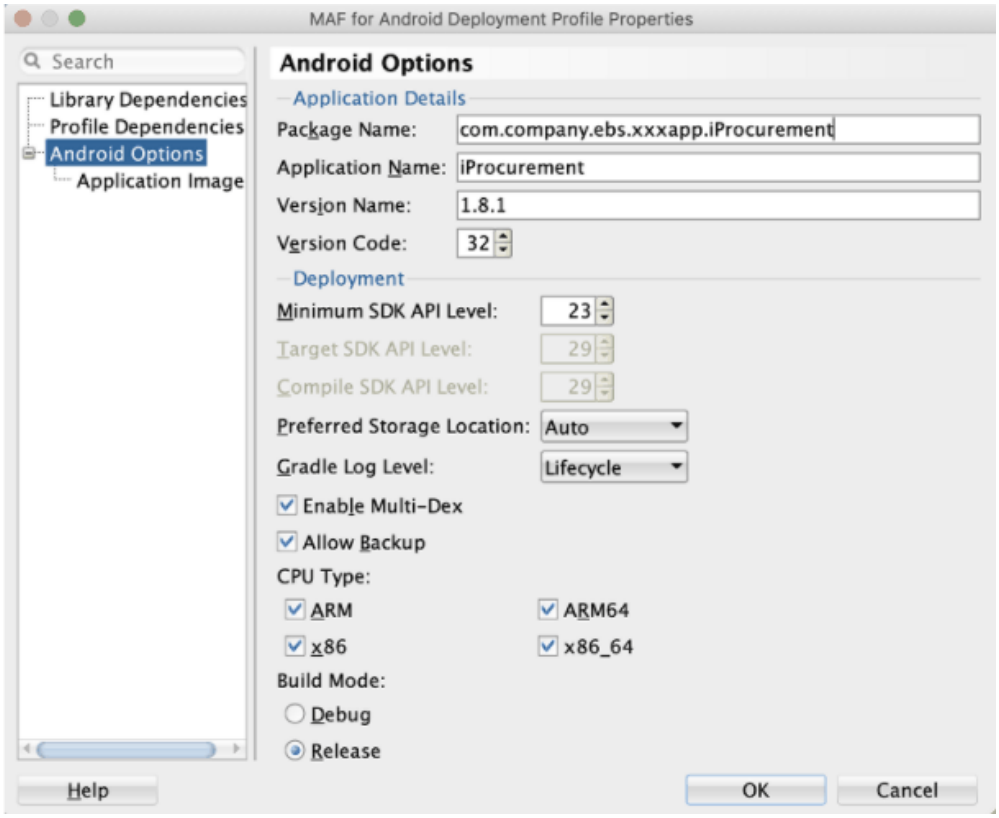
   Enter a new name, such as "XXX_iProcurement_Android", as the Deployment Profile Name if required.

Create Deployment Profile Page for Android



3. Enter the following information in the MAF for Android Deployment Profile Properties page:

MAF for Android Deployment Profile Properties Page



- Package Name: Enter a unique Id, such as `com.company.ebs.xxxapp.iProcurement` in this page. Use the same value from the Id field that you modified in the **`maf-application.xml`** file, as described earlier in the Changing Application Bundle Id, page 4-16.

  > **Note:** If the app supports push notifications, such as an enterprise-distributed Approvals app, the Package Name used here in the Android deployment profile is also used when creating the Android mobile client in Oracle Mobile Hub or Oracle Mobile Cloud Service. See: Creating Mobile Clients, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.
  >
  > For more information about push notifications, see Implementing Push Notifications, page 6-1.

- Application Name: This is the application that the package name belongs, such as `iProcurement`.

- Version Name: This is the release version of the application code that is displayed for the mobile users.
  - If you are customizing Oracle E-Business Suite mobile apps for enterprise distribution, retain the version.
  - If you create custom mobile apps, use your own app version mechanism, such as starting with 1.0.0.
- Version Code: Select an appropriate value from the drop-down list to represent the version of the application code. This code is checked programmatically by other applications for upgrades or downgrades. Please note that the minimum and default value is 1.

  This value is incremented by 1 for each subsequent release.
- Minimum SDK API Level: Select "23" from the drop-down list as the minimum SDK API level that the mobile app supports for the Android platform. This value 23 corresponds to the SDK API level for Android 6.0.

  Oracle E-Business Suite certifies mobile apps for platform version 6.0 (API Level 23) to 10.0 (API Level 29).
- Target SDK API Level: This value is defaulted based on the underlying Android SDK installation.
- Compile SDK API Level: This value is defaulted based on the underlying Android SDK installation.
- Build Mode: Select the **Debug** radio button for development and testing. Select the **Release** radio button for production deployment.

Click **OK**.

4. Save your work.

Changing the App Logo and Splash Screen

Once a new deployment profile is created, you can replace the default Oracle image used for the app icons with custom app logos or images.

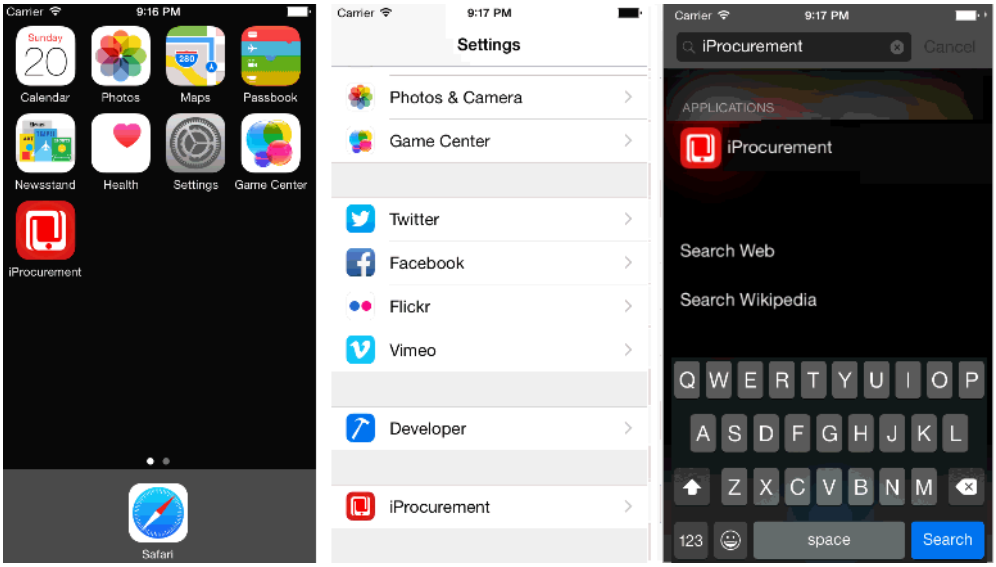
Customizing the App Logo for Corporate Branding

The app logo or icon is displayed in the About page. It can also be shown on an iOS device in the following places:

- A launcher icon on an iOS device's home screen (shown in the left)
- In the Settings screen against the app name (shown in the middle)

- In the Spotlight when the app is searched for (shown in the right)

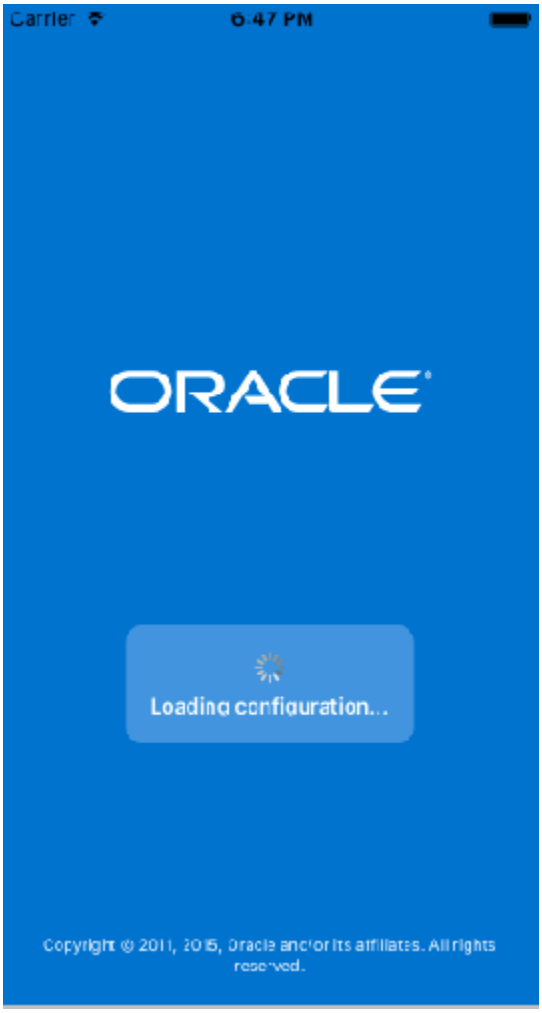
Customized App Logo in Home Screen, Settings Screen, and Spotlight Screen



Customizing Splash Screen for Corporate Branding

Splash screen is the first screen mobile users will see when launching an app.

Customized Splash Screen



This section includes the following topics:

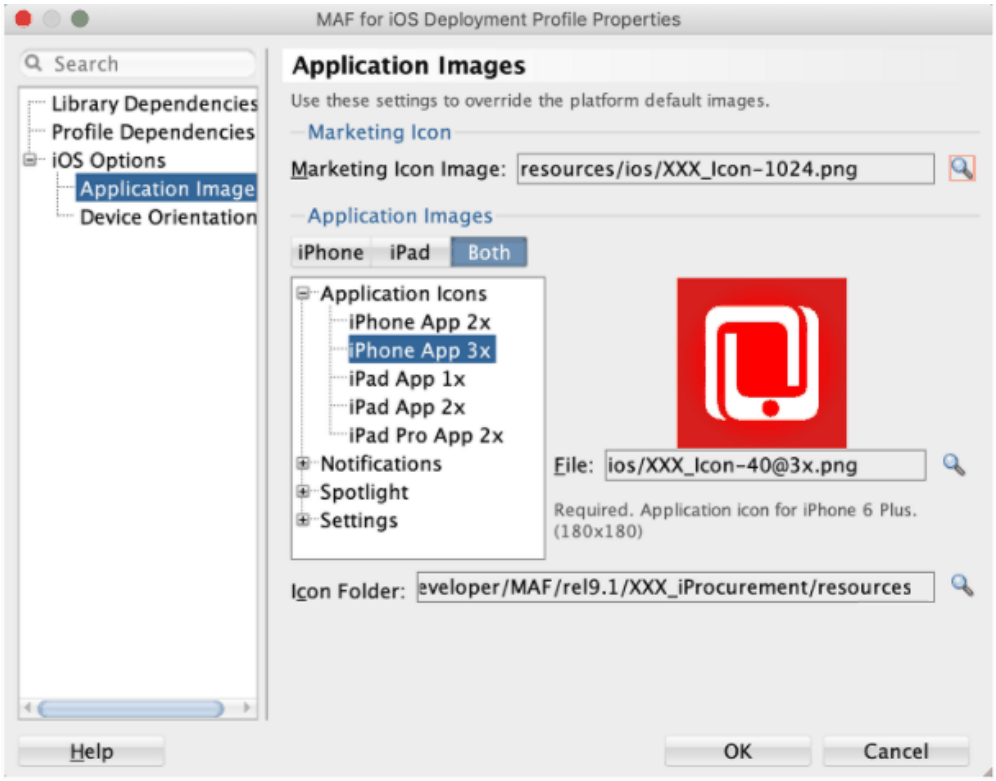
- Instructions to Change the App Logo for the iOS Platform, page 9-21
- Instructions to Change the Splash Screen for the iOS Platform, page 9-23
- Instructions to Change the App Logo and Splash Screen for the Android Platform, page 9-23

Instructions to Change the App Logo for the iOS Platform

Use the following steps to change the mobile app images for corporate branding:

1. In the MAF for iOS Deployment Profile Properties page, select "Application Images" from the iOS Options tree node.

Application Images Page for iOS Deployment Profile



2. In the Application Images region, select each device type tab if desired. Then choose an appropriate icon for the following categories, based on the image sizing tables as described in Required Image Sizing Information for the iOS Platform, page 9-4:

   Note: All images must be in `.png` format.

   - Marketing Icon
   - Application Icons
   - Notifications
   - Spotlight
   - Settings

3. Click **OK** to save the changes.

Instructions to Change the Splash Screen for the iOS Platform

If you implement corporate branding for an enterprise-distributed Oracle E-Business Suite mobile app that is modified from the associated MAA file, use the default Oracle E-Business Suite launch screen that is available under `ApplicationController/public_html/resources/html/ebs-ios-launch-screen.html` to further customize the app for corporate branding.

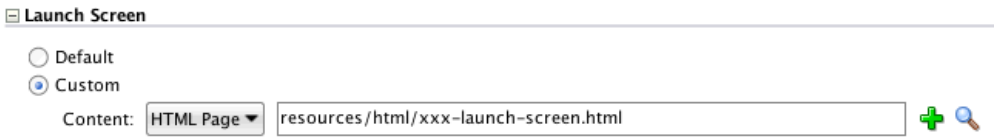
If you develop a custom mobile app, use the default MAF launch screen and customize it based on the MAF documentation.

For information on setting up the launch screens, see Configuring the Content of a MAF Application, *Developing Mobile Applications with Oracle Mobile Application Framework*.

Use the following steps to configure a custom iOS launch screen:

- Place the custom HTML-based iOS splash screen under your application's `Application Controller/public_html/resources/html` directory.
- Open your application's **`maf-application.xml`**. In the Application tab and the Launch Screen region, select the "Custom" radio button. Choose "HTML Page" as the content and browse to locate your custom splash screen.

Application Tab: Launch Screen Region

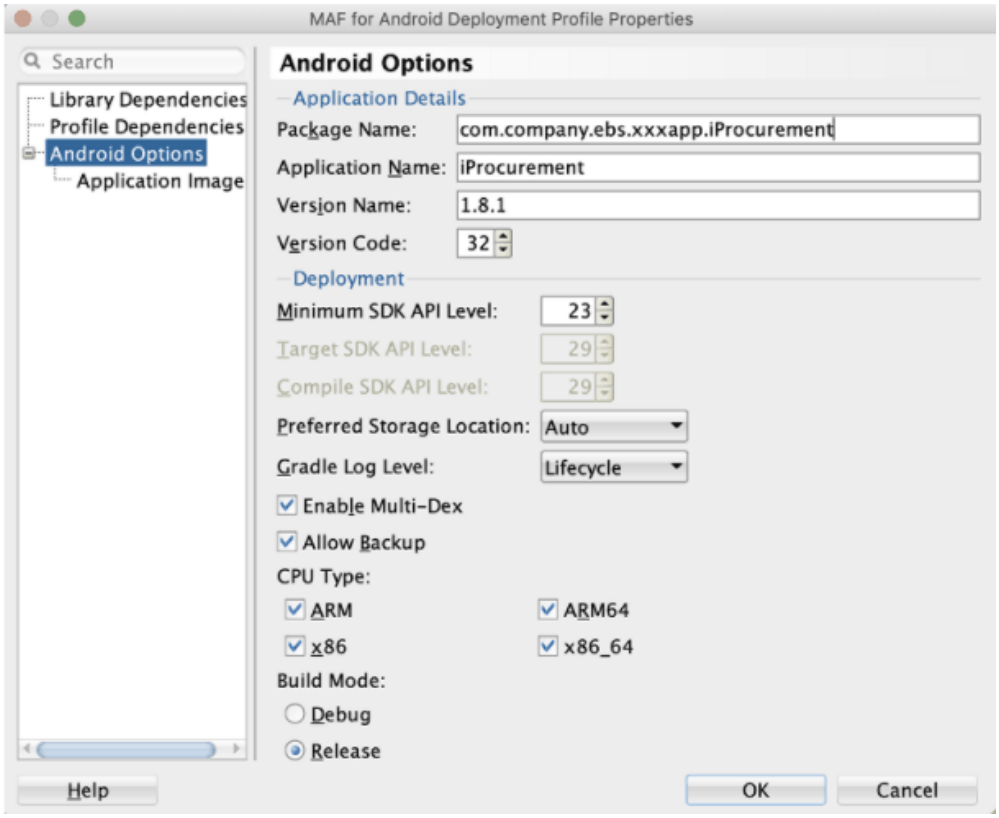


Instructions to Change the App Logo and Splash Screen for the Android Platform

Perform the following steps to change the app images for the Android platform:

1. In the MAF for Android Deployment Profile Properties page, select the "Android Options" tree node and enter the following information:

MAF for Android Deployment Profile Properties



- Package Name: Enter a unique package name for the mobile app on Android, such as `com.company.ebs.xxxapp.iProcurement`. Each app deployed to an Android device must have a unique package name, and this package name cannot start with a numeric value. Follow the same bundle Id naming conventions as described earlier in step 3, Instructions to Create a New Deployment Profile with Customized App Logos for the iOS Platform, page 9-11.

  Please note that the Id value specified in the configuration file **`maf-application.xml`** can be used for the deployment profile's package name. For information on defining the Id in the configuration file, see: Changing Application Bundle Id, page 4-16.

- Application Name: Enter the name of the .apk file created by Oracle MAF if needed. For example, XXXiProcurement.

- Version Name: The release version of the application code that is displayed for the mobile users. For more information about this field, see step 3 in Instructions to Create a New Deployment Profile with Customized App Logos for the Android Platform, page 9-16.

- Version Code: Select an appropriate value from the drop-down list to represent the version of the application code. This code is checked programmatically by other applications for upgrades or downgrades.

  Please note that the minimum and default value is 1. This value is incremented by 1 for each subsequent release.

Click **OK** to save your work.

2. Select "Application Images" from the tree node. In the Platform Display Types field, select each display type if appropriate and then choose an appropriate image for the Application Icon, Splash Screen (Portrait), and Splash Screen (Landscape) fields.

   For the required image sizing information and file names, see Required Image Sizing Information for the Android Platform, page 9-7.

   Note: All images must be in `.png` format.

   Click **OK** to save the changes.

Changing the App Logo in the About Page and Android Page Header

The app logo within an app is shown in the following places:

- About page for the iOS and Android devices (size 152x152)

  See: Displaying the App Logo in the About Page, page 9-25.

- Page header for the Android device (size 80x80)

  See: Displaying the App Logo in the Page Header for Android, page 9-26.

Both images should be rounded corners and in `.png` format.

Displaying the App Logo in the About Page

Perform the following steps to display your custom app logo in the About page for the iOS and Android platforms:

1. Copy the `.png` file whose image size is 152x152 to the `<ApplicationRootFolder>/.adf/META-INF/ebs/custom` directory.

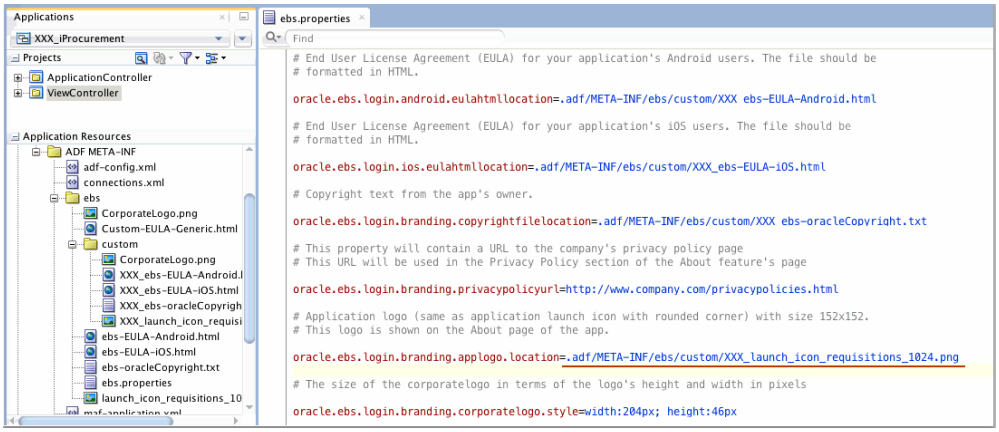
2. In Oracle JDeveloper, expand the **Application Resources** panel, then the **Descriptors** folder, then the **ADF META-INF** folder, and then the **ebs** folder.

   Double click the **`ebs.properties`** file.

3. Update the `oracle.ebs.login.branding.applogo.location` property with the following information:

```
oracle.ebs.login.branding.applogo.location=.adf/META-
INF/ebs/custom/XXX_launch_icon_requisitions_1024.png
```

App Logo Property in ebs.properties



Note that you can use any file name for the image file with size 152x152 as long as you refer to the correct name in this property.

Displaying the App Logo in the Page Header for Android

Note: This section is not required for the apps built with Oracle E-Business Suite Mobile Foundation Release 5.0.

Perform the following steps to display your custom app logo in the page header for Android apps:

1. Name the `.png` file whose image size is 80x80 as `android_app_header_icon.png`.

   It is important to note that the image file name should be exactly as mentioned here to display the app logo in the page header.

2. Copy the `.png` file to the `<ApplicationRootFolder>/ViewController/public_html/resources/images` directory.

3. If you have not performed the tasks to display the app logo in the About page as described earlier, you need to update the `oracle.ebs.login.branding.applogo.location` property in the **`ebs.properties`** file by performing the step 2 and step 3, as described in Displaying the App Logo in the About Page, page 9-25.

Implementing Your Company Logo

Mobile users can find a company logo in the following places within an app:

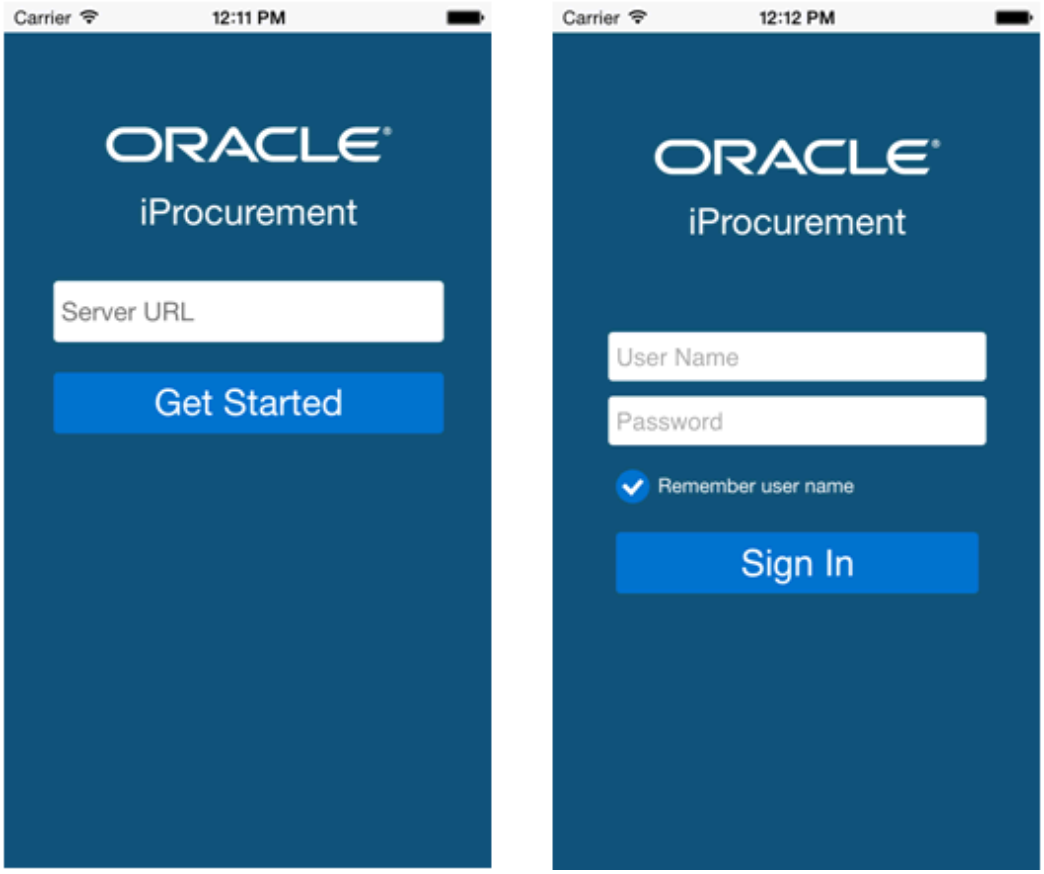
- Server URL screen (shown in the left)

  The Server URL screen is displayed only during the initial launch of an app or when an app is reconfigured.

- Sign In screen (shown in the right)

  Note: There is no company logo shown in the Sign In screen for Oracle Mobile Supply Chain for Oracle E-Business Suite (MSCA) that is not developed based Oracle E-Business Suite Mobile Foundation.

Company Logo in the Server URL Screen and Sign In Screen



Instructions to Implement Your Company Logo:

Perform the following steps to replace Oracle logo with your company logo:

1. Create and name the image file name as `CorporateLogo.png` and copy it to the following locations in the Oracle JDeveloper application folder:

   > **Note:** The image file name should be `CorporateLogo.png`. For the required sizing information, see Required Image Sizing Information for Corporate Branding, page 9-4.

   - `<ApplicationRootFolder>/.adf/META-INF/ebs/custom`
   - `<ApplicationRootFolder>/ApplicationController/public_html/resources/images`

2. In Oracle JDeveloper, expand the **Application Resources** panel, then the **Descriptors** folder, then the **ADF META-INF** folder, and then the **ebs** folder.

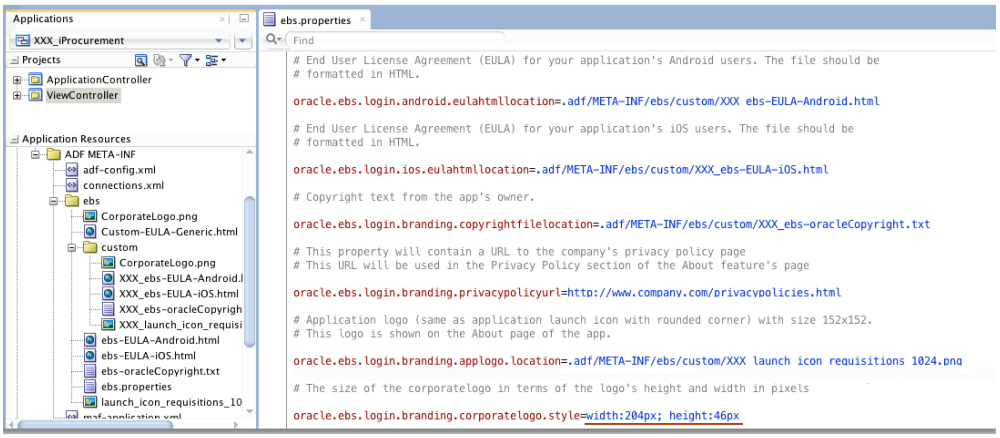
   Double click the **`ebs.properties`** file.

3. Update the following information in the **ebs.properties** file to change the company logo in the Server URL screen:

   oracle.ebs.login.branding.corporatelogo.style=width:204px; height:46px

   - This property lets you define an appropriate CSS style for your company logo.
   - The width and height in pixels should be 50% of the actual size. For example, if your image is 100x100, the CSS style should be `width=50px;height=50px;`.

Corporate Logo Property in ebs.properties



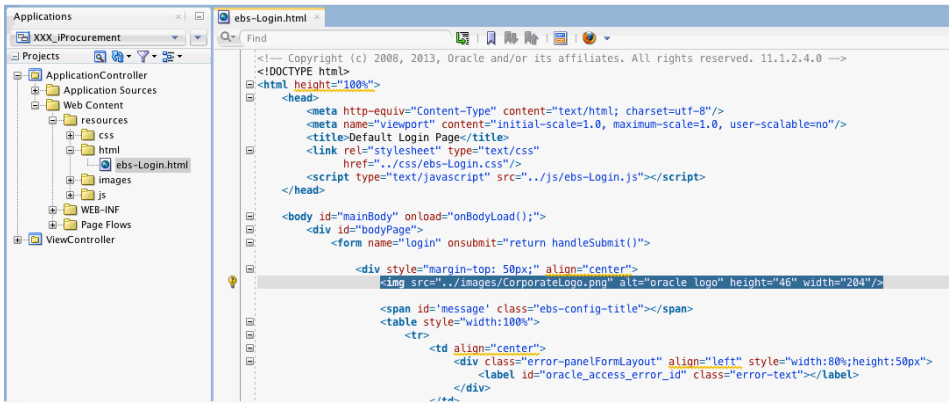
4. The image is automatically shown in the Sign In screen based on the file name `CorporateLogo.png`.

   Use the following steps to customize the CSS style so that the size of the logo is displayed correctly:

   1. Expand the **ApplicationController** folder, then the **Web Content** folder, then the **resources** folder, and then the **html** folder.
   2. Double click the **ebs-Login.html** file and then select the Source tab
   3. Search and locate the file name "`CorporateLogo.png`" in the following HTML tag:

      ```
      <img src="../images/CorporateLogo.png" alt="oracle logo"
      height="46" weight="204"/>
      ```

Company Logo CSS Style in ebs-Login.html



4. Update the height and width attributes with the same values used to update the property `oracle.ebs.login.branding.corporatelogo.style` as described in step 3.

Changing the App Name (Optional)

A mobile user can find the app name in many places when using or launching a mobile app. In addition to using your company logo for corporate branding, you can use app name of your choice in the following screens:

- Server URL screen
- Sign In screen
- Springboard
- About page

The app name shown in the Sign In screen comes from a specific file which is different from the rest of the screens in the list mentioned earlier.

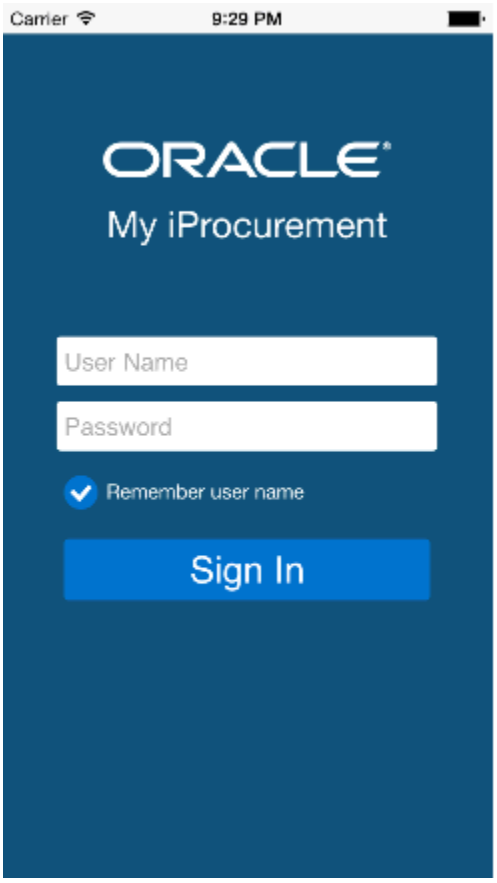
- For instructions on changing the app name in the Sign In screen, see Instructions to Use Your App Name in the Sign In Screen, page 9-30.
- For instructions on changing the app name in the Server URL screen, Springboard, and the About page, see Instructions to Use Your App Name in the Sign In Other Screens, page 9-33.

Instructions to Use Your App Name in the Sign In Screen

The following screenshot represents the app name change (from "iProcurement" to "My

iProcurement") shown in the Sign In screen.

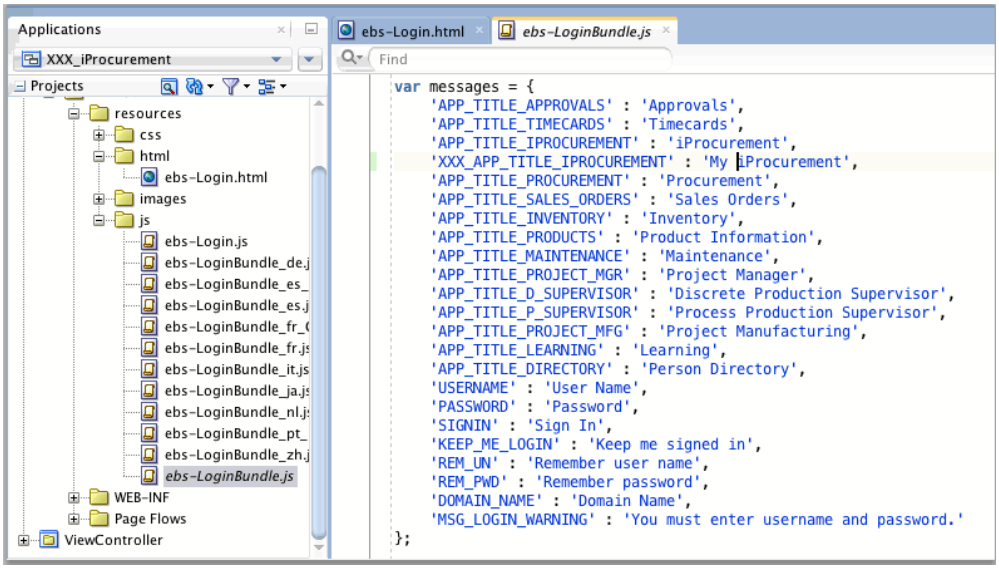
Desired App Name in the Sign In Screen



Perform the following steps to use your app name in the Sign In screen:

1. In Oracle JDeveloper, select your mobile application project.
2. Expand the **ApplicationController** folder, then the **Web Content** folder, then the **resources** folder, and then the **js** folder.
3. Double click the **`ebs-LoginBundle.js`** file.
4. Add a string in the **`ebs-LoginBundle.js`** file for your customized app name. For example, `'XXX_APP_TITLE_IPROCUREMENT' : 'My iProcurement',`.

Desired App Name in the ebs-LoginBundle.js File



- Please note that **`ebs-LoginBundle.js`** is a JSON file. Except for the last key-value pair, all pairs should end with a comma.

- This **`ebs-LoginBundle.js`** file can be translated to different languages if required. Place the translated files under the following folder for the mobile app:

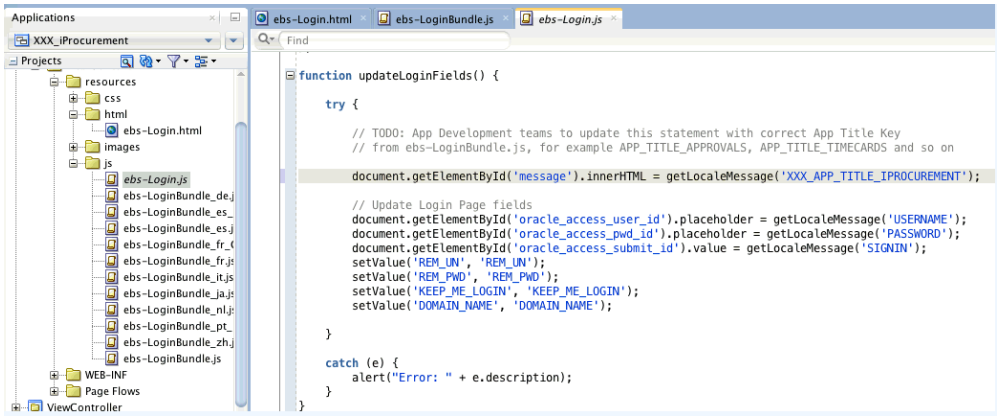
  `<ApplicationRootFolder>/ApplicationController/public_html/resources/js/`

Save your work.

5. Double click the **`ebs-Login.js`** file.

6. Replace the existing app name with your desired app name in the **`ebs-Login.js`** file.

Desired App Name in the ebs-Login.js File



Search for a string starting with `APP_TITLE` and replace the full string with your new app name key. For example, replace `APP_TITLE_IPROCUREMENT` with `XXX_APP_TITLE_IPROCUREMENT`.

7. Save your work.

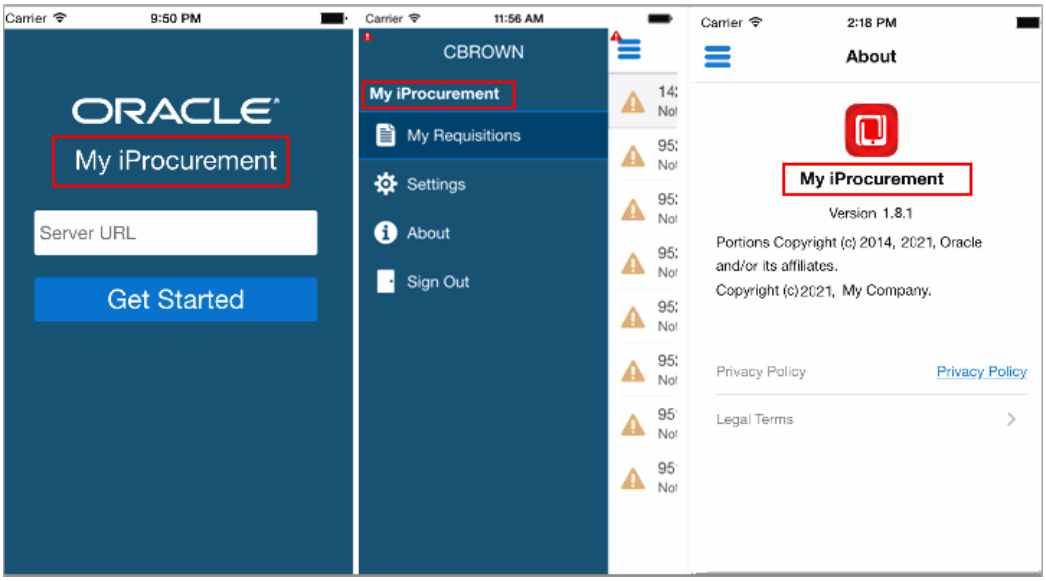
Instructions to Use Your App Name in Other Screens

In addition to the Sign In screen, your app name can be displayed in the Server URL screen, Springboard, and the About page.

For example, the app name is changed from "iProcurement" to "My iProcurement" in the Server URL screen (shown in the left) and the About page (shown in the right).

The changed app name "My iProcurement" is displayed at the top of the Springboard (shown in the middle) for corporate branding.

Desired App Name in the Server URL Screen, Springboard, and About Page



Instructions to Use Your App Name in Other Screens

Perform the following steps to change the app name in the Server URL screen, Springboard, and the About page:

1. Open Oracle JDeveloper.

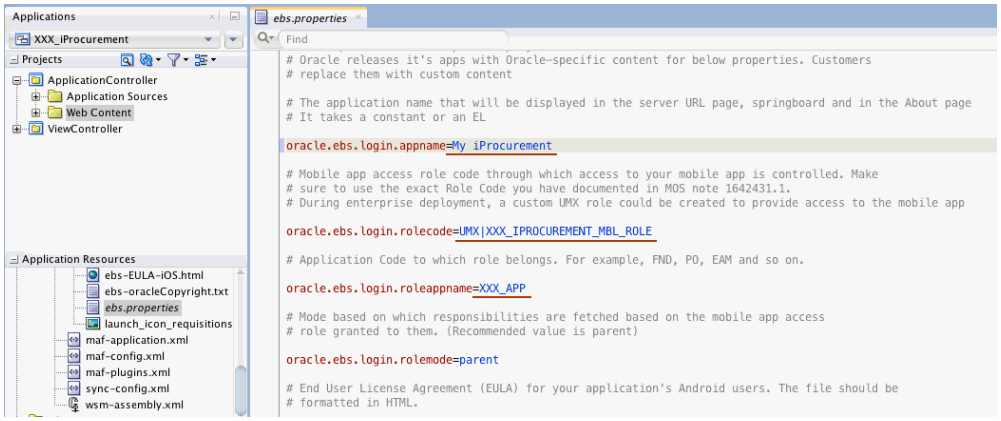
2. In the Application Navigator, expand the **Application Resources** panel, then the **Descriptors** folder, then the **ADF META-INF** folder, and then the **ebs** folder.

   Double click the **`ebs.properties`** file.

3. Update the following property in the **`ebs.properties`** file:

   ```
   oracle.ebs.login.appname=My iProcurement
   ```

App Name Property in the ebs.properties File



Ensure that your app name (`My iProcurement`) matches exactly what is used in the Sign In screen described earlier.

4. Save your work.

Customizing the Legal Related Information (Optional)

In addition to changing the company logo, app logo, splash screen, and app name, you can optionally customize legal related information specifically for your company. This includes:

- End User License Agreement (EULA) or Legal Terms

  This information is displayed during the initial launch of an app. For information on customizing the EULA, see Customizing End User License Agreement or Legal Terms (Optional), page 9-35.

- Copyright in the About Page

  Oracle allows you to modify the copyright information for corporate branding by optionally adding your own copyright information along with Oracle's copyright.

  See: Customizing Copyright in the About Page (Optional), page 9-37.

Customizing End User License Agreement or Legal Terms (Optional)

When an app is launched for the first time, the End User License Agreement (EULA) or Legal Terms screen appears. The app users need to accept the license agreement in order to use the app.

The app users can also find the same content through the About page by tapping the **Legal Terms** link. This content is stored in the following directory:

`<ApplicationRootFolder>/.adf/META-INF/ebs`

To display the customized End User License Agreement (EULA) or Legal Terms, perform the following steps:

1. Copy a generic file `Custom-EULA-Generic.html` that can be customized to the following folder:

   `<ApplicationRootFolder>/.adf/META-INF/ebs/custom`

2. Modify the generic `Custom-EULA-Generic.html` file to include the EULA content specific to your company.

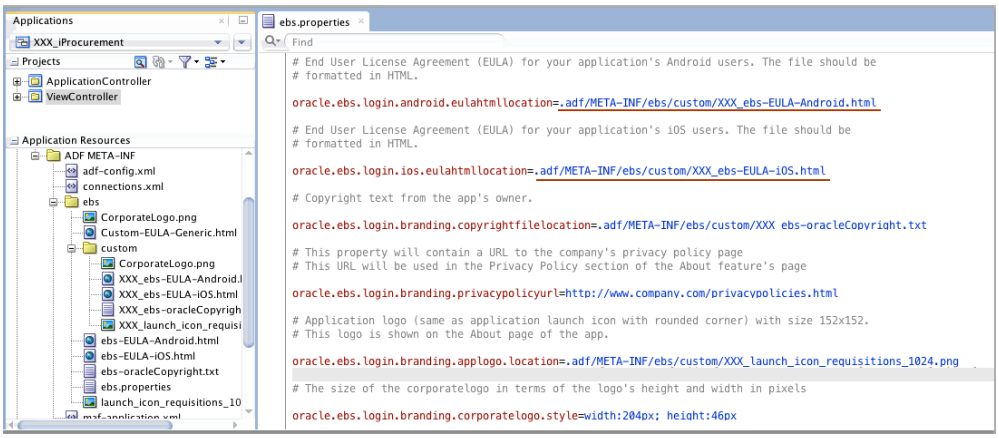
3. Rename the customized file, such as `XXX_ebs-EULA-Android.html` if it is for Android and `XXX_ebs-EULA-iOS.html` if it is for iOS.

   If the EULA content is the same for both iOS and Android, you could have the content in one file and update both properties with the same file.

4. After creating the customized EULA file, update the following properties in the **`ebs.properties`** file:

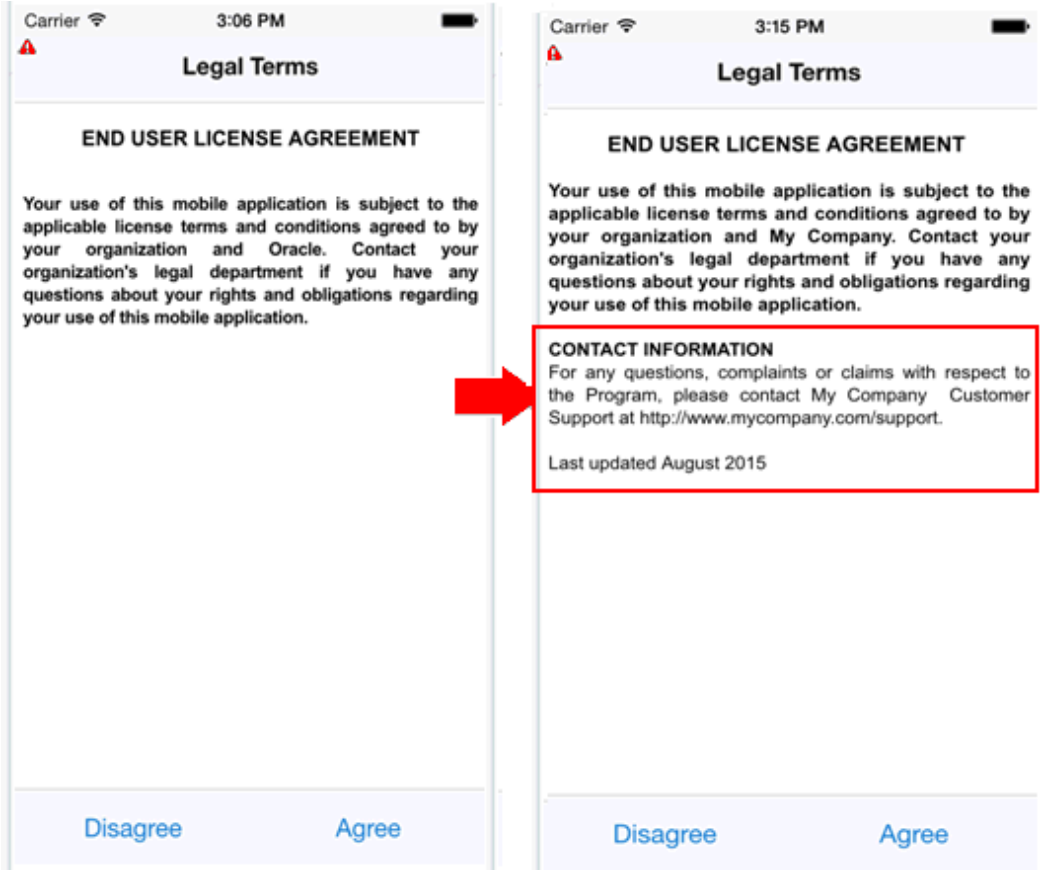
   - **`oracle.ebs.login.android.eulahtmllocation`**`=.adf/META-INF/ebs/custom/XXX_ebs-EULA-Android.html`
   - **`oracle.ebs.login.ios.eulahtmllocation`**`=.adf/META-INF/ebs/custom/XXX_ebs-EULA-iOS.html`

End User License Agreement Properties in the ebs.properties File



After the modification, when the app is launched for the first time, the customized End User License Agreement content appears on an iOS device as shown this example.

Legal Terms Screen (Left) with Customized End User License Agreement Highlighted (Right)



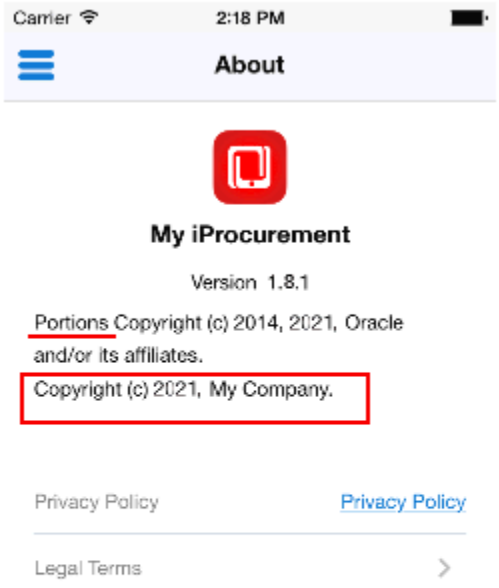
Customizing Copyright in the About Page (Optional)

Copyright information is displayed in the About page which is accessible from the Springboard of an Oracle E-Business Suite mobile app by tapping the **About** link.

Information displayed in the About page, such as copyright, privacy policy URL, is specific to Oracle E-Business Suite mobile apps. In addition to the required privacy policy change as described earlier for enterprise distribution, you can modify the copyright information for corporate branding by optionally adding your company's own copyright information along with Oracle's copyright.

In this example, the app logo, company logo, and copyright text have been modified specifically for corporate branding, along with the privacy policy URL change.

About Page with Customized Copyright Highlighted



- For information on updating the app name, see Changing the App Name, page 9-30 .
- For information on changing privacy policy URL, see Changing the Privacy Policy Link, page 4-19.

Instructions to Customize the Copyright Information in the About page

Perform the following steps to customize the copyright information:

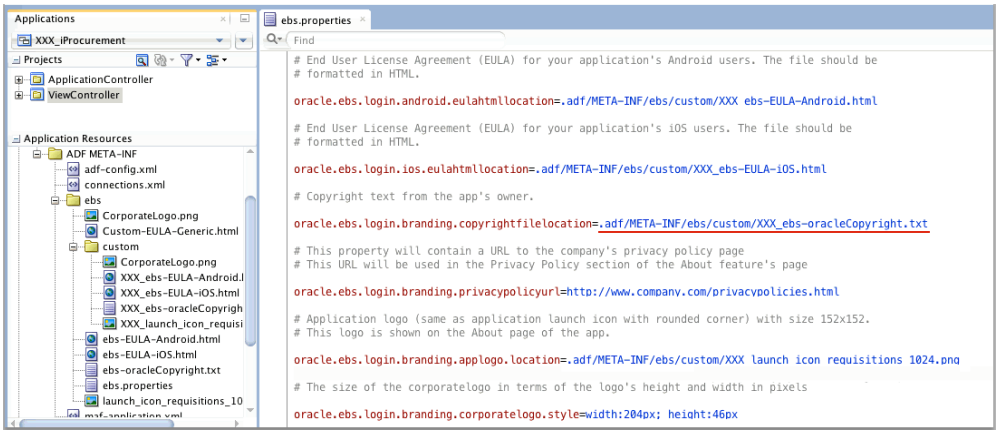
1. Open Oracle JDeveloper.
2. Create a file `<ApplicationRootFolder>/.adf/META-INF/ebs/custom/XXX_ebs-oracleCopyright.txt` and include your own copyright information if needed.
3. In the Application Navigator, expand the **Application Resources** panel, then the **Descriptors** folder, then the **ADF META-INF** folder, and then the **ebs** folder.

   Double click the **`ebs.properties`** file.
4. Update the copyright information in the **`ebs.properties`** file:

oracle.ebs.login.branding.copyrightfilelocation=.adf/META-INF/ebs/custom/XXX_ebs-oracleCopyright.txt

Note: Leave the Oracle copyright information unchanged in the About page and it should not be removed.

Copyright Property in the ebs.properties File



5. Save your work.

Deploying Your Apps

Deploying the Generated Application Binary Files

After completing the required changes, you can deploy the app for the iOS, Android, or both platforms. At this stage, you have already performed either one of the following tasks for the deployment profile(s) depending on whether it is an enterprise-distributed Oracle E-Business Suite mobile app or a custom mobile app.

- Modified an existing deployment profile to change only the Application Bundle Id for enterprise distribution

  See: Modifying an Existing Deployment Profile (Conditional), page 4-23.

- Created a new deployment profile with customized app images for corporate branding

  See: Creating a New Deployment Profile with Customized App Logos and Splash Screen, page 9-11.

Use the appropriate deployment profile to deploy and test the app in the iOS simulators or Android emulators. For information on setting up the iOS and Android simulators, see: Installing Development Tools and Setting Up a Development Environment, page 3-23.

If the test results are successful, you can generate the application binary file. For iOS, the archive format is an IPA file, known as an iOS application bundle. For Android, the format is an Android application package (APK) file.

> **Important:** Enterprise-distributed mobile apps must be distributed to an enterprise's own site and are only available to the enterprise's internal users. These apps cannot be distributed to any third party users. Additionally, they cannot be redistributed to a public app store.

For more information on deploying the app to an iOS simulator or an Android emulator as well as deploying the app to an iOS-powered or Android-powered device, see Deploying MAF Applications, *Developing Mobile Applications with Oracle Mobile Application Framework*.

A

Oracle E-Business Suite Mobile APIs in the Oracle Integration Repository

Overview

This appendix lists the published Oracle E-Business Suite mobile APIs that are available in the Oracle Integration Repository where you can search, view, and deploy them as REST services for your custom app development.

For information on using these APIs as REST services, refer to the following section based on your mobile release:

- For Oracle E-Business Suite Mobile Apps Release 10.x, see Oracle E-Business Suite APIs, page 1-2.
- For Oracle E-Business Suite Mobile Apps Release 9.x and Earlier, see Implementing Oracle E-Business Suite REST Services, page 7-4.

Oracle E-Business Suite Mobile APIs Available in Oracle Integration Repository

Mobile App Name	Interface Name	Operation Name
Oracle Mobile Approvals for Oracle E-Business Suite	Approvals Management	• Get Open Notifications • Get Pending Approvals By Approval Type • Get Approval Types

Mobile App Name	Interface Name	Operation Name
	Approvals Details and Actions	• Get Approval Header • Get Approval Details • Submit Approval • Reassign Approval • Get Approval Participants • Request Information • Provide Information
Oracle Mobile Timecards for Oracle E-Business Suite	Timecard Information and Actions	• Submit Timecard • Delete Timecard • Get Day Details • Get Timecard Status • Get Timecard Details

Mobile App Name	Interface Name	Operation Name
Oracle Mobile Learning for Oracle E-Business Suite	Learning Details	• Get Current Learning Details • Get Certification Member Details • Get Learning Path Details • Get Learning Path Courses of a Section • Search Learning Path • Search Courses • Search Certifications • Search Classes • Get Root Category Details • Get Certification Details • Get Course Details • Get Class Details • Get Browse Catalog Details • View Review Ratings • Get Category Image Details • Get Class Session Details

Mobile App Name	Interface Name	Operation Name
	Learning Actions	• Initialize Learner Details • Save Review Rating • Enroll to Class • Subscribe to Certification • Subscribe to Learning Path • Unsubscribe From Learning Path • Move Learning Path To History • Unsubscribe From Certification • Move Certification to History • Renew Certification • Move Class to History • Unenroll From Class • Get Class Enrollment Warning Messages
Oracle Mobile Person Directory for Oracle E-Business Suite	Person Directory	• Get Person Directory Details • Get Person Reportees Details

Mobile App Name	Interface Name	Operation Name
Oracle Mobile iProcurement for Oracle E-Business Suite	iProcurement Mobile Services	• Get My Requisitions • Get PO Details • Get Receipt Details • Get Requisition Approval History • Get Requisition Details • Get Requisition Line Details • Get Requisition Line Shipments • Get Requisition Lines getReqLines
Oracle Mobile Procurement for Oracle E-Business Suite	Procurement Mobile Services	• Get Order Details • Get PO Action History • Get PO Distributions • Get PO Lines • Get PO Orders • Get PO Shipments

Mobile App Name	Interface Name	Operation Name
Oracle Mobile Project Manager for Oracle E-Business Suite	Project Management	• Get Change Order List • Get Customer Invoice Details • Get Customer Invoice List • Get Issue Action List • Get Issue List • Get Project Customer Details • Get Project Customer List • Get Project Details • Get Project List • Get Supplier Invoice Details • Get Supplier Invoice List • Get Team Members List

Mobile App Name	Interface Name	Operation Name
Oracle Mobile Discrete Production Supervisor for Oracle E-Business Suite	WIP ISG Mobile REST Services	• Accounting Class LOV • Assign Shop Floor Status • Department LOV • Organization LOV • Shop Floor Status LOV • Work Order Action • Work Order Details • Work Order List • Work Order Move Op • Work Order Reschedule Job
Oracle Mobile Discrete Quality Manager for Oracle E-Business Suite	Discrete Quality Manager Collection Plans	• Get Child Plan Results • Get Child Plans List • Get Parent Plan for Child • Get Work Order Plans List
	Discrete Quality Manager Work Orders	• Get List of Work Orders • Get Operations Details • Get Work Order Summary • Get Work order Operations List

Mobile App Name	Interface Name	Operation Name
Oracle Mobile Inventory for Oracle E-Business Suite	Cycle Count Manual Schedule	• Cycle Count Manual Schedule creation

Mobile App Name	Interface Name	Operation Name
Oracle Mobile Process Production Supervisor for Oracle E-Business Suite	GME Batch/Step Services	• Get Responsibility Organization List • Get Batch Summary Count • Get Starting Today Batch List • Get Completing Today Batch List • Get On-Track Batch List • Get Delayed Batch List • Get Batches with exceptions • Get Open Batch List • Get Batch Detail • Get Batch Steps • Get Batch Sales Orders • Get Batch Expiring Lots • Get Batch Unallocated Ingrdient List • Get Batch Mateirals • Release Batch • Complete Batch • Terminate Batch • Cancel Batch • Reschedule Batch

Mobile App Name	Interface Name	Operation Name
		• Get Step Summary Counts • Get Step Detail • Get Batch Step Resource List • Get Step Expiring Lot Exceptions • Get Step Unallocated Material Exceptions • Get Step Associated Items • Reschedule Step • Get Open Step List • Get Starting Today Steps • Get Completing Today Steps • Get On Track Steps • Get Delayed Steps • Get Steps with Exceptions

Mobile App Name	Interface Name	Operation Name
Oracle Mobile Process Quality Manager for Oracle E-Business Suite	GMD Sample Services	• Get Lov Action Control Attributes • Get Sup Sample More Details • Get Quality Samples Details • Change Planned Disposition • Get Not Assigned Samples List • Change Disposition • Get Reason Codes • Get Lot Status Codes • Get Out Of Spec Samples List • Get Approval Samples List • Get Lot Grades • Get Cust Sample More Details • Get Delayed Samples List • Get Samples List • Get Analysis Samples List • Get Wip Samples Cord Info • Get Inv Samples Cord

Mobile App Name	Interface Name	Operation Name
		Info • Get Sup Samples Cord Info • Get Due Today Samples List • Get Collection Samples List • Get Cus Samples Cord Info • Get Wip Sample More Details • Get Inv Sample More Details • Get Disposition Codes

Mobile App Name	Interface Name	Operation Name
	GMD Sample Test Services	• Get Unassigned Sample Test List • Get Delayed Sample Test List • Get OutofSpec Sample Test List • Get Due Today Sample Test List • Get Pending Evaluation Test List • Get Sample Tests • Evaluate Test • Update Sample Test Tester/Planned Resource • Get Sample Tests • Get Sample Test Detail • Get Sample Test count • Get Pending Result Sample Test List • Get Responsibility Organization List

Mobile App Name	Interface Name	Operation Name
Oracle Mobile Product Information for Oracle E-Business Suite	Product Information	• Get Organizations • Get Favorite Items • Get Items • Change Item Favorite • Get Item Overview • Get Attribute Groups • Get Attribute Group Detail
Oracle Mobile Project Manufacturing for Oracle E-Business Suite	Project Manufacturing Services	• Get Borrow Details • Get Borrow Summary • Get On-hand Quantities • Get Payback Details • Get Project Materials

Mobile App Name	Interface Name	Operation Name
Oracle Mobile Sales Orders for Oracle E-Business Suite	Order Management Sales Orders	• Get Additional Order Line Details • Get All Order Lines Details • Get All Orders Related to Bill to Customers • Get Distinct Bill to Customers • Get Hold And Backordered Orders Related to Bill to Customers • Get Hold and Backordered Lines Details • Get Invoice Details • Get Invoice Total Details • Get Line Invoice Details • Get Order Line Details • Get Sales Order Details • Get Sales Orders

Mobile App Name	Interface Name	Operation Name
Oracle Mobile Self-Service Human Resources for Oracle E-Business Suite	Self-Service HR	• Get Oracle SSHR Person Details • Get Person Directs Details • Get Assignment Data • Get Person Absence Type Balances • Get Person Recent Absences List • Get Person Planned Absences List • Get Current User Details and Preferences • Calculate Absence Duration
	Change Pay Details	• Get Change Pay Current Salary Details • Get Employee Grade Details • Get Change Pay Salary Proposal Reason List • Get Change Pay Pending Proposal Transaction Details

Mobile App Name	Interface Name	Operation Name
	Payslip	• Get List of Available Payslips • Get detailed payslip information • Get Payslip PDF blob as encoded String
	US Tax Form Details	• Get Person Tax Record information
	Year End Form Details (US Localization)	• Get W2 Forms Information
	Canadian Year End Form Details	• Get T4 Forms Information • Get T4A Forms Information • Get RL1 Forms Information • Get RL2 Forms Information
	Pay Simulator	• Submit Simulation Run • Get Simulation Results
	India Tax Reports	• Get Reports List • Get Form16 Report • Get Form12BB Report • Get PDF Report

Mobile App Name	Interface Name	Operation Name
Oracle Mobile Yard for Oracle E-Business Suite	Yard Appointment Details	• Get Dock Appointment List • Get Dock Appointment Details
	Yard Statistics Details	• Get Yard Statistics
	Accessible Yard Organizations	• Get YMS Organizations

B

Using the Sample App as a Reference

Overview

Oracle E-Business Suite mobile Login component provides a sample mobile app that lets you use as a reference implementation. It implements various features of the Login component that are described earlier for Oracle E-Business Suite Mobile Apps Release 9.x and Earlier only. It does not apply to Oracle E-Business Suite Mobile Apps Release 10.x.

To better understand the sample app, this appendix includes the following topics:

Sample App Features

The sample app implements the following features in Oracle E-Business Suite as a sample only for developers to use as a reference. This app should not be deployed against a production environment.

- **Notifications**

  This feature provides the ability to view open notifications in the current user's worklist. This feature is not the same as in the Approvals app (Oracle Mobile Approvals for Oracle E-Business Suite). This is an implementation that uses its own REST service implementation to show the list of all OPEN notifications in the user's worklist. This feature requires the following two pages:

  - Notifications.amx

    This page invokes the Application Module REST service

`/webservices/rest/mobileamsample/notifications`.

- Details.amx

  This page invokes the PL/SQL REST service `/webservices/rest/mobileplsqlsample/notificationdetails`.

- **Status Monitor**

  This feature provides the ability to search for workflow processes on Oracle E-Business Suite and review the status of the activities within the process. This feature requires the following pages:

  - SMSearch.amx

    This is a search page that a user enters search criteria for workflow processes.

  - WFTypeLOV.amx

    This page implements a LOV (List of Values) to search for workflow item types. This LOV invokes the Application Module REST service `/webservices/rest/mobileamsample/wfTypes`. The LOV is invoked from the search page.

  - SMList.amx

    This page displays a list of workflow items that match the search criteria. For each workflow item, the status is also shown. This page invokes the Application Module REST service `/webservices/rest/mobileamsample/wfItems`.

  - SMDetails.amx

    This page displays the complete list of runtime activities within a workflow item and the status of each in the order of their implemenation. This page invokes the Application Module REST service `/webservices/rest/mobileamsample/itemActivities`.

- **Settings**

  This feature is implemented as part of the sample app. However, this page only shows how you can integrate your app-specific Settings screen with the Login component's default features, such as responsibility picker, connection details, and diagnostics.

- **About**

  The About page is part of the Login component. You need to add the About feature to Springboard and it is not required to make any changes.

- **Sign Out**

The Sign Out feature is part of the Login component. Similar to the About feature, you need to add the Sign Out feature to Springboard and it is not required to make any changes.

Sample App Patterns

Apart from providing several built-in features through the Login component, the sample app implements the following patterns that you can refer to and use in your mobile apps:

- Simple search
- Advanced search
- Pagination
- LOV (List of Values)
- Messages
- Barcode scanner

Preparing to Run the Sample App

The sample app includes the following components:

- REST APIs that are required by the sample app to fetch data from Oracle E-Business Suite
- Oracle Mobile Application Framework project of the sample app that uses the REST APIs to fetch data from Oracle E-Business Suite

Before you run the sample app, ensure that you perform the following tasks:

1. Enable the REST interfaces used by the sample app through the following steps:
    1. Ensure that Oracle E-Business Suite Integrated SOA Gateway (ISG) is set up correctly and the REST service framework is enabled, as described in Configuring Oracle E-Business Suite REST Services, page 7-3 and Implementing Oracle E-Business Suite REST Services, page 7-4.

        Important: If the REST service framework is not enabled, patches for the sample app REST services cannot be applied successfully.

Note: Configuring Oracle E-Business Suite REST services provided through ISG is also required if you use Oracle E-Business Suite REST APIs for custom app development. However, it is not required if you use standard Oracle E-Business Suite mobile apps installed from the Apple App Store or Google Play, or apps provided to users through enterprise distribution.

2. Ensure you have applied the following patches for the REST services that the sample app uses:

   Note: These two patches are included in all product family patches which are consolidated into the Oracle E-Business Suite level patches. See: Applying Patches for Mobile Apps Built with Oracle E-Business Suite Mobile Foundation Release 9.1 and Release 9.0, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and Release 12.2*.

   - For Oracle E-Business Suite Release 12.1.3, apply patch 21316087:R12.FND.B.
   - For Oracle E-Business Suite Release 12.2, apply patch 21316087:R12.FND.C.

   These patches contain the following information:

   - Sample REST service implementations
   - Application definition metadata for the sample app
   - A mobile app access role for the sample app

3. Log in to Oracle E-Business Suite as a user who has the Integration Administrator role (or the Integration Repository Administrator role in Oracle E-Business Suite 12.1.3 and 12.2.3). Select the **Integrated SOA Gateway** responsibility and the **Integration Repository** link from the navigation menu.
4. Select "Product Family" from the View By drop-down list in the Integration Repository.
5. Expand the "Applications Technology" folder, then the "Application Object Library" folder, and then select "Mobile Sample".

   This retrieves the following APIs that the sample app uses.

   - Mobile Sample AM Interface

- Mobile Sample PL/SQL Interface
- Organization Picker

6. For each interface that the sample app uses, perform the following tasks:
   1. Select the Grants tab.

      Create security grants for all the operations contained in the selected interface so that the users who will be running the sample app can access these operations.

      For more information on creating security grants, refer to Managing Grants for Interfaces with Support for SOAP and REST Web Services, Administering Native Services, *Oracle E-Business Suite Integrated SOA Gateway Implementation Guide*.

   2. Select the REST Web Service tab.

      Enter the following information for each interface:

      - **Service Alias:** Specify service alias information for each selected interface:
        - Mobile Sample AM Interface - mobileamsample
        - Mobile Sample PL/SQL Interface - mobileplsqlsample
        - Organization Picker - mobileorgsample
      - **Verb:** Select "POST" as the HTTP verb for each selected interface.

   3. Click **Deploy** to deploy each selected interface as a REST service.

      For more information about deploying REST services, refer to Chapter 3 Administering Native Services, *Oracle E-Business Suite Integrated SOA Gateway Implementation Guide*.

7. The REST interfaces are now ready to be used by the sample mobile app.

2. Enable the sample app by configuring the app in Oracle E-Business Suite:
   1. Log in to Oracle E-Business Suite as a user who has the **Mobile Applications Manager** responsibility.

      Select the **Mobile Applications Manager** responsibility and choose the **Applications** link from the navigation menu.

   2. Locate the sample mobile app in the Search Mobile Applications page by

entering the following information:

- **Application Name:** EBS Sample App
- **Application Bundle ID:** com.oracle.ebs.atg.mbl.Sample

3. Enable and configure the sample app by clicking the **Configure** icon.
4. If desired, specify configuration parameter values for the sample app.

For information on configuring a mobile app, refer to Configuring the Mobile Apps on the Oracle E-Business Suite Server, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

3. Assign the sample app access role (role code `UMX|FND_MBL_SAMPLE_ROLE`) to responsibilities through the following steps:
   1. Identify one or two responsibilities that are already available to the Oracle E-Business Suite users using which you plan to log in to the sample app.
   2. Assign the sample app access role to those responsibilities.

   For information on assigning the role to responsibilities, refer to Setting Up Mobile App Access to Responsibilities, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

Deploying and Running the Sample App

Unzip the sample app and open it in Oracle JDeveloper that is already set up to run Oracle Mobile Application Framework applications. Deploy the sample app to either an iOS Simulator or Android Emulator and test it.

Please note that the Developer mode is disabled for the sample app by default. Therefore, when you run it, you are required to accept the End User License Agreement and enter the Server URL for the Oracle E-Business Suite environment where you have enabled the REST APIs and configured the app.

C

Supported Properties in the Login Component

Supported Properties

The following table lists all the properties in the `ebs.properties` file and how each property can be used to implement different features of the Login component described earlier for Oracle E-Business Suite Mobile Apps Release 9.x and Earlier only. It does not apply to Oracle E-Business Suite Mobile Apps Release 10.x.

> **Important:** If you have already copied and modified the `ebs.properties` file from a previous version of the Login component, you need to manually merge the new `ebs.properties` file into your existing file and configure the new properties appropriately.

Property Name	Description	Possible Values
oracle.ebs.login.server.url (available from Oracle E-Business Suite Mobile Foundation Release 8.0 and onwards)	This is the Oracle E-Business Suite server URL that an app will use to connect to Oracle E-Business Suite by default. If a valid Oracle E-Business Suite server URL is entered for this property, the app users will not be prompted to enter a Server URL when the app is launched for the first time.	A valid Oracle E-Business Suite Server URL, such as `https://example.com`

Property Name	Description	Possible Values
oracle.ebs.login.server.url.allow_change (available from Oracle E-Business Suite Mobile Foundation Release 8.0 and onwards)	If a default Oracle E-Business Suite server URL is used in property `oracle.ebs.login.server.url`, this property indicates whether mobile users are allowed to change it in an app. By default, mobile users are not allowed to change the server URL. Change its value to **Y** only if you allow users to change the default server URL.	Y or N
oracle.ebs.login.useresppicker	• **N:** Do not use the responsibility picker in the app. Your app does not require application responsibility context to retrieve data from Oracle E-Business Suite. • **Y:** Use the responsibility picker in the app. Your app requires application responsibility context. This will display the responsibility picker after a user signs in to the app for the first time.	Y or N

Property Name	Description	Possible Values
oracle.ebs.login.init.defaultfeature	If oracle.ebs.login.useresppicker=Y, this property should point to the MAF feature Id of the page for any additional steps in the context initialization flow after a user picks a responsibility in the Responsibility Picker screen. For example, if the organization picker is required for your app after a responsibility is selected, then the value of this property would be the feature Id of the organization picker. This property is applicable only when oracle.ebs.login.useresppicker=Y.	Valid MAF Feature Id
oracle.ebs.login.responsibilitypicker.usedonebutton	• **Y:** Show the **Done** button on the Responsibility Picker screen. This marks the end of the context initialization flow and after a user taps the **Done** button, the Login component shows the app's landing page. • **N:** Show the **Next** button on the Responsibility Picker screen. This means that a user can continue to the next step in the context initialization flow and the app displays the feature set in the property oracle.ebs.login.init.defaultfeature. This property is applicable only when oracle.ebs.login.useresppicker=Y.	Y or N

Property Name	Description	Possible Values
oracle.ebs.login.initializeresp	• **Y:** The responsibility context is initialized when a user leaves the Responsibility Picker screen (either using the **Done** or **Next** button) and the selected responsibility is stored in the local database. • **N:** The selected responsibility is set to application scope variables but the context is not initialized. Developers should read the selected responsibility value from the application scope variables, store it in the local database using the provided API and initialize required context. This scenario is useful when the app uses the organization picker immediately following the responsibility picker. This gives developers the control on when the security context is initialized. This property is applicable only when the oracle.ebs.login. useresppicker=Y.	Y or N

Property Name	Description	Possible Values
oracle.ebs.login.app.defaultfeature	This is the MAF feature Id of the app's functional default landing page after the application context is initialized. This is the first feature that a user will see after the user logs in once the initial setup is completed. This property value can also be updated from the application using the API `PreferenceStore.setDefaultFeature (String userName, String featureId);`	Valid MAF Feature Id
oracle.ebs.login.dbname	Do not use. Oracle Internal use only.	
oracle.ebs.login.dbkey	Do not use. Oracle Internal use only.	
oracle.ebs.login.sql	This is the path to a SQL file that contains required DML statements to create a local database on the device. This database is specific a mobile app in which the app data can be stored on the device. The `.sql` file should be placed inside the `.adf/META-INF/ebs` folder. For example, `.adf/META-INF/ebs/myapp.sql`.	Path to SQL file
oracle.ebs.features.settings.id	This is the MAF feature Id of the app's Settings feature that is accessible from Springboard. This property is required when oracle.ebs.login.useresppicker=Y.	Valid MAF Feature Id of the Settings feature

Property Name	Description	Possible Values
oracle.ebs.login.rolecode	This is the internal code of the mobile app access role that is granted to certain responsibilities that can access this mobile app. If your app requires the responsibility context (oracle.ebs.login.useresppicker=Y), it is required that you create a role using the User Management responsibility, grant that role to the responsibilities that can access this app. The same role code should be set up in this property.	A valid UMX role code
oracle.ebs.login.roleappname	Application Code to which the mobile app access role belongs. For example, FND, PO, EAM and so on.	App short name
oracle.ebs.login.rolemode	Leave the default value "parent".	parent
oracle.ebs.login.springboard.type	Use List only. Grid is not supported.	List, Grid
oracle.ebs.login.android.eulahtmllocation	This is the location of End User License Agreement (EULA) for your app's Android users and is placed under `.adf/META-INF/ebs`. The file should be formatted in HTML. For example, `.adf/META-INF/ebs/Android-EULA.html`.	Valid HTML file path

Property Name	Description	Possible Values
oracle.ebs.login.ios.eulahtmllocation	This is the location of End User License Agreement (EULA) for your app's iOS users and is placed under `.adf/META-INF/ebs`. The file should be formatted in HTML. For example, `.adf/META-INF/ebs/iOS-EULA.html`.	Valid HTML file path
oracle.ebs.login.branding.showeula	Enable or disable the feature of showing the EULA content. • **Y:** Show the EULA content when the app is launched for the first time and in the About screen. • **N:** Hide the EULA content when the app is launched for the first time and in the About screen.	Y or N
oracle.ebs.login.branding.copyrightfilelocation	This is the location of a text file that contains any copyright content for the app. This property is required only if the app has any copyright text in the About screen of the app. For example, `.adf/META-INF/ebs/My-Copyright.txt`.	Valid text file path
oracle.ebs.login.branding.privacypolicyurl	URL to the company's privacy policy page to be shown to the users in the About screen of the app.	Valid URL

Property Name	Description	Possible Values
oracle.ebs.login.branding.applogo.location	App logo (same as the app launch icon with rounded corner) with size 152x152. This logo is shown on the About screen of the app. For example, `.adf/META-INF/ebs/my_app_icon.png`.	PNG image file location
oracle.ebs.login.bundle.name	Full XLF bundle name in which your app's translatable app name is stored. If your app's name is "Product Information", you should define the value in the ViewController project bundle and include its path here.	A bundle with a fully qualified name
oracle.ebs.login.appname	App name to show on the Server URL, Springboard and About screens. This property takes the following values: • Actual value of the app, such as "Approvals", "Product Information" and so on. Such values are shown on the app as is and not translated. • EL expression that evaluates to a translated string from the XLF bundle setup mentioned earlier for the property oracle.ebs.login.bundle.name.	An EL or a constant

Property Name	Description	Possible Values
oracle.ebs.login.branding.corporatelogo.style	Styling for the corporate logo to be shown in the Server URL screen. This property takes the height and width styles for the corporate logo. The height and width should be specified in pixels. The height and width values should be one half of the actual height and width of the image. For example, if a corporate logo image is 408 by 92 pixels, then the value of this property would be `width:204px; height: 46px`.	Height and width style values in pixels
oracle.ebs.login.development.mode	• **Y:** Run the app in the developer mode. The app does not show the EULA and Server URL screens. • **N:** Default value if this property is not set. The app runs in the production mode.	Y or N
oracle.ebs.login.development.server.url	Oracle E-Business Suite server URL that an app will connect to. This property is used only if the developer mode is turned on.	Valid Oracle E-Business Suite server URL
oracle.ebs.login.development.logging.finest	• **Y:** Turn on FINE Logging for the app. • **N:** Default value if this property is not set. This property is used only if the developer mode is turned on.	Y or N

D

Mobile Application Archives for Earlier Oracle E-Business Suite Mobile Foundation Releases

Overview

This appendix lists the Oracle E-Business Suite Mobile Application Archives for the most recent releases prior to the current release, Oracle E-Business Suite Mobile Foundation 9.1. This information includes the software distribution part numbers that you can use to download the mobile application archive files from the Oracle Software Delivery Cloud (https://edelivery.oracle.com/) page and the MAA file details for the associated Oracle E-Business Suite mobile apps. Specifically, it includes the following Oracle E-Business Suite Mobile Foundation releases:

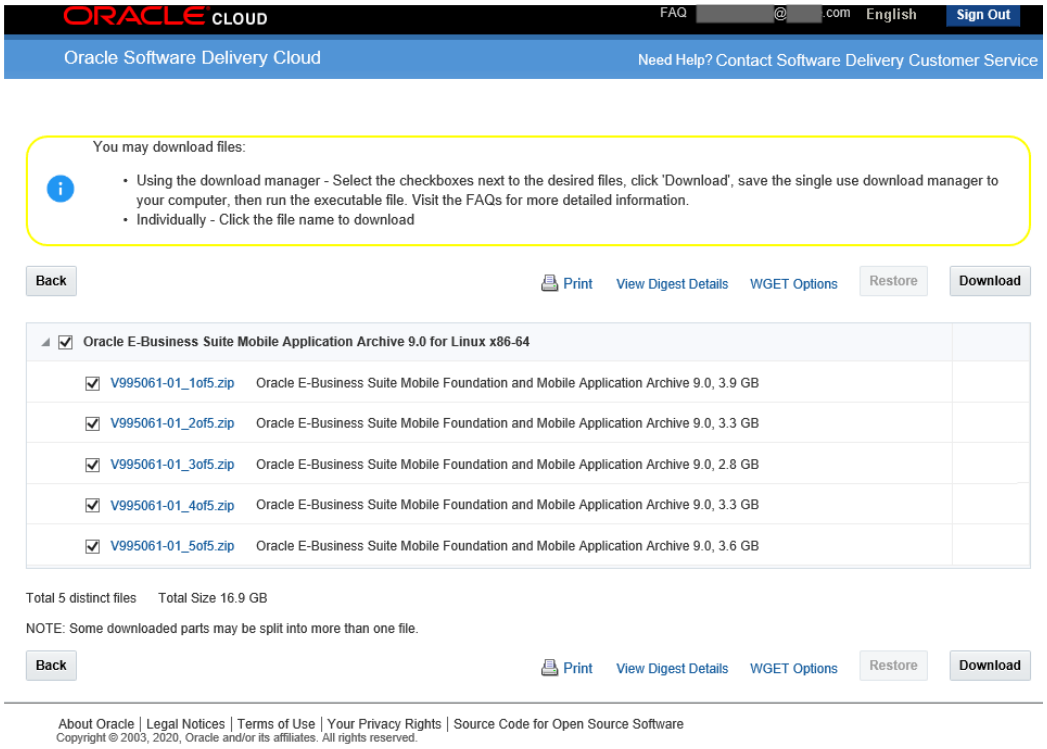
Mobile Application Archives for Oracle E-Business Suite Mobile Foundation Release 9.0

This "Oracle E-Business Suite Mobile Application Archive 9.0" software distribution (`V995061-01`) is divided into the following five files. You should download all these five files to obtain the needed components for enterprise distribution and custom app development.

Note: This software distribution also contains the Mobile Application Archive file for Oracle Mobile Supply Chain Applications for Oracle E-Business Suite (MSCA), although this app is not developed based on Oracle E-Business Suite Mobile Foundation.

- `V995061-01_1of5.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 9.0
- `V995061-01_2of5.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 9.0
- `V995061-01_3of5.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 9.0
- `V995061-01_4of5.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 9.0
- `V995061-01_5of5.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 9.0

Oracle Software Delivery Cloud Page with Oracle E-Business Suite Mobile Application Archive 9.0 Software Distribution



This software distribution contains the following components:

- Consolidated server-side prerequisite patches

  For information about these server-side patches for the apps developed based on Oracle E-Business Suite Mobile Foundation and Oracle Mobile Supply Chain Applications for Oracle E-Business Suite (MSCA), see the *Oracle E-Business Suite Mobile Foundation and Mobile Application Archive Release 9.0 Readme*, included in this software distribution. Additionally, see Applying Prerequisite Patches on the Oracle E-Business Suite Server, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

- Oracle Mobile Application Framework 2.6.2 for Oracle E-Business Suite Mobile Foundation Release 9.0

  This Oracle MAF version containing additional bug fixes on top of Oracle Mobile Application Framework 2.6.2 is specifically built for Oracle E-Business Suite Mobile Foundation Release 9.0. To work with downloaded Mobile Application Archive files, you need to use this specific Oracle MAF version that is included in this software distribution.

- Oracle E-Business Suite Mobile Foundation package, for enabling custom app development for Oracle E-Business Suite

For information about the Oracle E-Business Suite Mobile Foundation package and how to use it to develop custom apps, see Using the Login Component to Develop Mobile Apps, page 5-1.

- Mobile Application Archive (.maa) files delivered in zip files, along with the app-specific readme for each Oracle E-Business Suite mobile app included in this software distribution

  You can use the app-specific MAA file to customize the app for enterprise distribution and corporate branding, and then distribute the updated version of the app to your users through your enterprise's own site rather than a public app store.

  - The following table lists the MAA file information associated with each Oracle E-Business Suite mobile app:

    Important: Oracle E-Business Suite mobile apps may republish updated MAA files after a general availability release if necessary. Before downloading the MAA files listed in the following table from this software distribution, you should review the corresponding product release notes first to check for any updated MAA information. For information about product release notes, refer to My Oracle Support Knowledge Document 1641772.1, *Oracle E-Business Suite Mobile Apps, Release 12.1 and 12.2 Documentation Index*.

Oracle E-Business Suite Mobile Application Archive Files for Oracle E-Business Suite Mobile Apps

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile Approvals for Oracle E-Business Suite (**Footnote 4**, page D-5)	Patch 30769811	p30769811_R12_GENERIC.zip
Oracle Mobile Discrete Production Supervisor for Oracle E-Business Suite	Patch 30850730	p30850730_R12_GENERIC.zip
Oracle Mobile Inventory for Oracle E-Business Suite	Patch 30797624	p30797624_R12_GENERIC.zip
Oracle Mobile Learning for Oracle E-Business Suite	Patch 30769494	p30769494_R12_GENERIC.zip

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile Maintenance for Oracle E-Business Suite	Patch 30806039	p30806039_R12_GENERIC.zip
Oracle Mobile Person Directory for Oracle E-Business Suite	Patch 30761720	p30761720 _R12_GENERIC.zip
Oracle Mobile Process Production Supervisor for Oracle E-Business Suite	Patch 30843433	p30843433_R12_GENERIC.zip
Oracle Mobile iProcurement for Oracle E-Business Suite	Patch 30844782	p30844782_R12_GENERIC.zip
Oracle Mobile Sales Orders for Oracle E-Business Suite	Patch 30770293	p30770293_R12_GENERIC.zip
Oracle Mobile Self-Service Human Resources for Oracle E-Business Suite	Patch 30761715	p30761715_R12_GENERIC.zip
Oracle Mobile Timecards for Oracle E-Business Suite	Patch 31303576	p31303576_R12_GENERIC.zip
Oracle Mobile Yard for Oracle E-Business Suite	Patch 30737757	p30737757_R12_GENERIC.zip
Oracle Mobile Supply Chain for Oracle E-Business Suite	Patch 30737734	p30737734_R12_GENERIC.zip
(**Footnote 5**, page D-5)		

Footnote 4: If you plan to distribute the Approvals app through enterprise distribution from the associated MAA file, you can optionally enable push notifications when using Oracle Mobile Hub or Oracle Mobile Cloud Service. For more details about implementing this feature for the mobile app, see Implementing Push Notifications, page 6-1.

Footnote 5: This software distribution also contains the MAA file for Oracle

Mobile Supply Chain for Oracle E-Business Suite, although this app is not built with Oracle E-Business Suite Mobile Foundation.

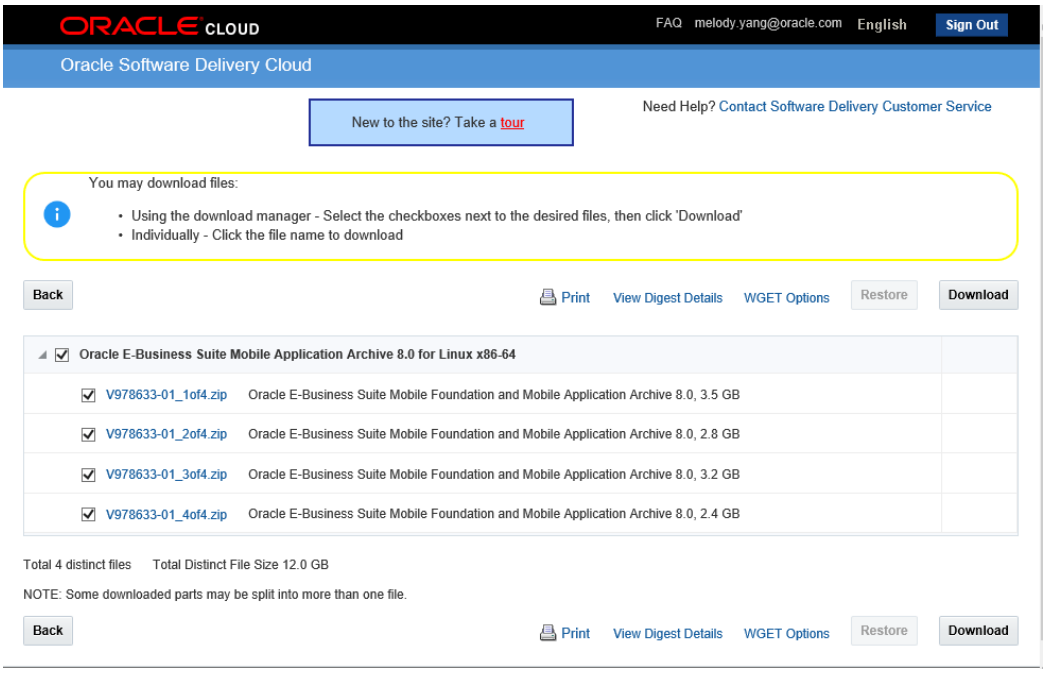
Mobile Application Archives for Oracle E-Business Suite Mobile Foundation Release 8.0

This "Oracle E-Business Suite Mobile Application Archive 8.0" software distribution (`V978633-01`) is divided into the following four files. You should download all these four files to obtain the needed components for enterprise distribution and custom app development.

> **Note:** This software distribution also contains the Mobile Application Archive file for Oracle Mobile Supply Chain Applications for Oracle E-Business Suite (MSCA), although this app is not developed based on Oracle E-Business Suite Mobile Foundation.

- `V978633-01_1of4.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 8.0
- `V978633-01_2of4.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 8.0
- `V978633-01_3of4.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 8.0
- `V978633-01_4of4.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 8.0

Oracle Software Delivery Cloud Page with Oracle E-Business Suite Mobile Application Archive 8.0 Software Distribution



This software distribution contains the following components:

- Consolidated server-side prerequisite patches

  For information about these server-side patches for the apps developed based on Oracle E-Business Suite Mobile Foundation and Oracle Mobile Supply Chain Applications for Oracle E-Business Suite (MSCA), see the *Oracle E-Business Suite Mobile Foundation and Mobile Application Archive Release 8.0 Readme*, included in this software distribution. Additionally, see Applying Prerequisite Patches on the Oracle E-Business Suite Server, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

- Oracle Mobile Application Framework 2.5.0 for Oracle E-Business Suite Mobile Foundation Release 8.0

  This Oracle MAF version containing additional bug fixes on top of Oracle Mobile Application Framework 2.5.0 is specifically built for Oracle E-Business Suite Mobile Foundation Release 8.0. To work with downloaded Mobile Application Archive files, you need to use this specific Oracle MAF version that is included in this software distribution.

- Oracle E-Business Suite Mobile Foundation package, for enabling custom app development for Oracle E-Business Suite

  For information about the Oracle E-Business Suite Mobile Foundation package and

how to use it to develop custom apps, see Using the Login Component to Develop Mobile Apps, page 5-1.

- Mobile Application Archive (.maa) files delivered in zip files, along with the app-specific readme for each Oracle E-Business Suite mobile app included in this software distribution

  You can use the app-specific MAA file to customize the app for enterprise distribution and corporate branding, and then distribute the updated version of the app to your users through your enterprise's own site rather than a public app store.

  - The following table lists the MAA file information associated with each Oracle E-Business Suite mobile app:

    Important: Oracle E-Business Suite mobile apps may republish updated MAA files after a general availability release if necessary. Before downloading the MAA files listed in the following table from this software distribution, you should review the corresponding product release notes first to check for any updated MAA information. For information about product release notes, refer to My Oracle Support Knowledge Document 1641772.1, *Oracle E-Business Suite Mobile Apps, Release 12.1 and 12.2 Documentation Index*.

Oracle E-Business Suite Mobile Application Archive Files for Oracle E-Business Suite Mobile Apps

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile Approvals for Oracle E-Business Suite (**Footnote 6**, page D-9)	Patch 27971128	p27971128_R12_GENERIC.zip
Oracle Mobile Discrete Production Supervisor for Oracle E-Business Suite	Patch 28213463	p28213463_R12_GENERIC.zip
Oracle Mobile Inventory for Oracle E-Business Suite	Patch 27809289	p27809289_R12_GENERIC.zip
Oracle Mobile Learning for Oracle E-Business Suite	Patch 28025110	p28025110_R12_GENERIC.zip

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile Maintenance for Oracle E-Business Suite	Patch 27835989	p27835989_R12_GENERIC.zip
Oracle Mobile Person Directory for Oracle E-Business Suite	Patch 28022354	p28022354_R12_GENERIC.zip
Oracle Mobile Process Production Supervisor for Oracle E-Business Suite	Patch 28023994	p28023994_R12_GENERIC.zip
Oracle Mobile iProcurement for Oracle E-Business Suite	Patch 28026707	p28026707_R12_GENERIC.zip
Oracle Mobile Sales Orders for Oracle E-Business Suite	Patch 27971016	p27971016_R12_GENERIC.zip
Oracle Mobile Self-Service Human Resources for Oracle E-Business Suite	Patch 28022406	p28022406_R12_GENERIC.zip
Oracle Mobile Timecards for Oracle E-Business Suite	Patch 28025556	p28025556_R12_GENERIC.zip
Oracle Mobile Yard for Oracle E-Business Suite	Patch 27953614	p27953614_R12_GENERIC.zip
Oracle Mobile Supply Chain Applications for Oracle E-Business Suite	Patch 27955047	p27955047_R12_GENERIC.zip
(**Footnote 7**, page D-9)		

Footnote 6: If you plan to distribute the Approvals app through enterprise distribution from the associated MAA file, you can optionally enable push notifications when using Oracle Mobile Cloud Service. For more details about implementing this feature for the mobile app, see Implementing Push Notifications, page 6-1.

Footnote 7: This software distribution also contains the MAA file for Oracle

Mobile Supply Chain Applications for Oracle E-Business Suite, although this app is not built with Oracle E-Business Suite Mobile Foundation.

Mobile Application Archives for Oracle E-Business Suite Mobile Foundation Release 7.0

This section describes the mobile application archive download information for the mobile apps that are built with Oracle E-Business Suite Mobile Foundation Release 7.0.

Downloading "Oracle E-Business Suite Mobile Application Archive 7.0" Software Distribution

Log in to Oracle Software Delivery Cloud (https://edelivery.oracle.com/) page and follow the download steps as described in Downloading Mobile Application Archives Files, page 4-5 to select "Oracle E-Business Suite Mobile Application Archive 7.0" (part number V861706-01) software distribution instead.

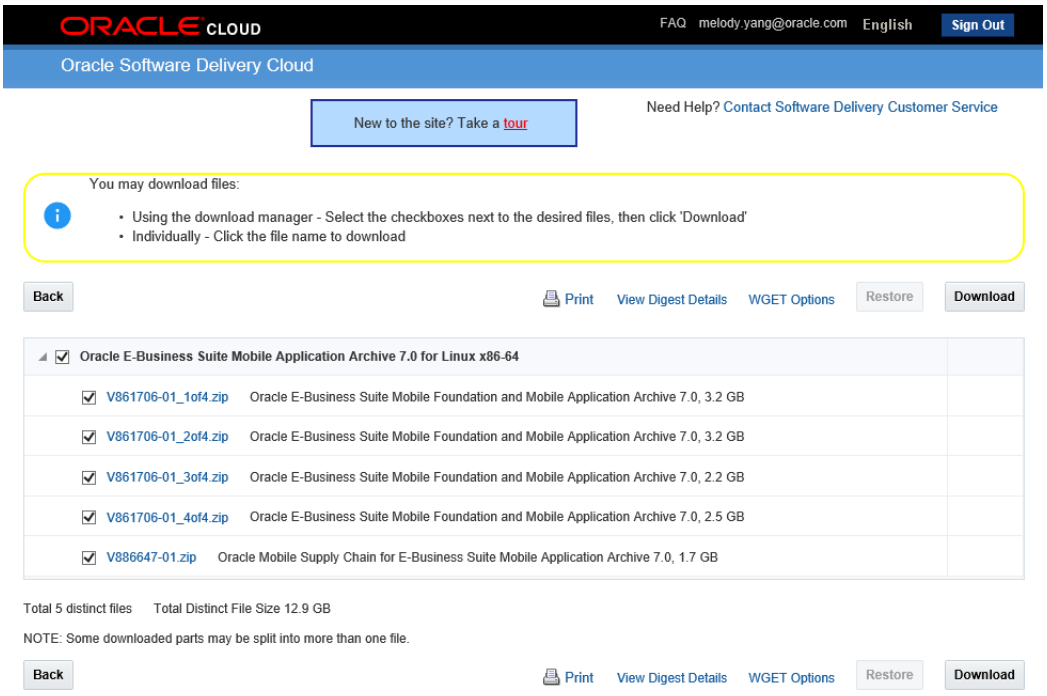
This software distribution includes the following four files for the apps built with Oracle E-Business Suite Mobile Foundation 7.0. You need to download these four files to obtain the needed components for enterprise distribution and custom app development:

- `V861706-01_1of4.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 7.0
- `V861706-01_2of4.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 7.0
- `V861706-01_3of4.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 7.0
- `V861706-01_4of4.zip` Oracle E-Business Suite Mobile Foundation and Mobile Application Archive 7.0

Additionally, this software distribution includes the following MSCA Mobile Application Archive 7.0 file for you to download through part number **`V886647-01`**:

- `V886647-01.zip` Oracle Mobile Supply Chain for E-Business Suite Mobile Application Archive 7.0

Oracle Software Delivery Cloud Page with Oracle E-Business Suite Mobile Application Archive 7.0 Software Distribution



This software distribution contains the following components:

- Consolidated server-side prerequisite patches

  For information about these server-side patches for the apps developed based on Oracle E-Business Suite Mobile Foundation, see the *Oracle E-Business Suite Mobile Foundation and Mobile Application Archive Release 7.0 Readme*, included in this software distribution. Additionally, see Product Family Patches for Oracle E-Business Suite Mobile Foundation Release 7.0, *Oracle E-Business Suite Mobile Apps Administrator's Guide, Release 12.1 and 12.2*.

- Oracle Mobile Application Framework 2.4.0 for Oracle E-Business Suite Mobile Foundation Release 7.0

  This Oracle MAF version containing additional bug fixes on top of Oracle Mobile Application Framework 2.4.0 is specifically built for Oracle E-Business Suite Mobile Foundation Release 7.0. To work with downloaded Mobile Application Archive files, you need to have this specific Oracle MAF version that is included in this software distribution.

- Oracle E-Business Suite Mobile Foundation package, for enabling custom app development for Oracle E-Business Suite

  For information about the Oracle E-Business Suite Mobile Foundation package and

how to use it to develop custom apps, see Using the Login Component to Develop Mobile Apps, page 5-1.

- Mobile Application Archive (.maa) files delivered in zip files, along with the app-specific readme for each Oracle E-Business Suite mobile app

The following table lists the MAA file information associated with each Oracle E-Business Suite mobile app built with Oracle E-Business Suite Mobile Foundation Release 7.0:

Oracle E-Business Suite Mobile Application Archive Files for Oracle E-Business Suite Mobile Apps Built with Oracle E-Business Suite Mobile Foundation 7.0

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile Approvals for Oracle E-Business Suite (**Footnote 8**, page D-13)	Patch 25974387	p25974387_R12_GENERIC.zip
Oracle Mobile Discrete Production Supervisor for Oracle E-Business Suite	Patch 25942813	p25942813_R12_GENERIC.zip
Oracle Mobile Discrete Quality Manager for Oracle E-Business Suite	Patch 26160018	p26160018_R12_GENERIC.zip
Oracle Mobile Inventory for Oracle E-Business Suite	Patch 25942708	p25942708_R12_GENERIC.zip
Oracle Mobile Learning for Oracle E-Business Suite	Patch 25945021	p25945021_R12_GENERIC.zip
Oracle Mobile Maintenance for Oracle E-Business Suite	Patch 25962228	p25962228_R12_GENERIC.zip
Oracle Mobile Person Directory for Oracle E-Business Suite	Patch 25947011	p25947011_R12_GENERIC.zip
Oracle Mobile Process Production Supervisor for Oracle E-Business Suite	Patch 25943517	p25943517_R12_GENERIC.zip

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile Process Quality Manager for Oracle E-Business Suite	Patch 26180810	p26180810_R12_GENERIC.zip
Oracle Mobile iProcurement for Oracle E-Business Suite	Patch 26197450	p26197450_R12_GENERIC.zip
Oracle Mobile Procurement for Oracle E-Business Suite	Patch 26197483	p26197483_R12_GENERIC.zip
Oracle Mobile Product Information for Oracle E-Business Suite	Patch 25949274	p25949274_R12_GENERIC.zip
Oracle Mobile Project Manager for Oracle E-Business Suite	Patch 25949676	p25949676_R12_GENERIC.zip
Oracle Mobile Project Manufacturing for Oracle E-Business Suite	Patch 25933918	p25933918_R12_GENERIC.zip
Oracle Mobile Sales Orders for Oracle E-Business Suite	Patch 25953213	p25953213_R12_GENERIC.zip
Oracle Mobile Self-Service Human Resources for Oracle E-Business Suite	Patch 25947023	p25947023_R12_GENERIC.zip
Oracle Mobile Timecards for Oracle E-Business Suite	Patch 25984682	p25984682_R12_GENERIC.zip
Oracle Mobile Yard for Oracle E-Business Suite	Patch 25943487	p25943487_R12_GENERIC.zip

Footnote 8: If you plan to distribute the Approvals app through enterprise distribution from the associated MAA file, you can optionally enable push notifications when using Oracle Mobile Cloud Service. For more details about implementing this feature for the mobile app, see Implementing Push Notifications, page 6-1.

> **Note:** For the MAA file associated with MSCA mobile app, you can download it through `V886647-01.zip` included in this software

distribution:

Oracle Mobile Supply Chain for Oracle E-Business Suite Mobile Application Archive File

Mobile App Name	Associated MAA Patch	Associated MAA File
Oracle Mobile Supply Chain Applications for Oracle E-Business Suite	Patch 26301633	p26301633_R12_GENERIC.zip

Index

G

I

M

O

P

R

S

T

U

W