Oracle® Mobile Application Framework

Installing Oracle Mobile Application Framework 2.1.0 E59150-01

January 2015

Documentation that describes how to install the Oracle Mobile Application Framework for use with Oracle JDeveloper to create mobile applications that run natively on devices.



Oracle Mobile Application Framework Installing Oracle Mobile Application Framework 2.1.0

E59150-01

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Primary Authors: Liza Rekadze, Walter Egan

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

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

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Pr	eface		v				
	Audien	ce	. v				
	Docum	entation Accessibility	. v				
	Related Documents						
	Conver	itions	. v				
1	Installi	ng Mobile Application Framework with JDeveloper					
	1.1	Introduction to Installing the MAF Extension with JDeveloper	1-1				
	1.2	Installation Requirements for MAF Applications to be Deployed to the iOS Platform	1-2				
	1.3	Installation Requirements for MAF Applications to be Deployed to the Android Platfor 1-2	rm				
	1.4	Setting Up JDeveloper	1-3				
	1.5	Installing the MAF Extension in JDeveloper	1-4				
2	Setting	JUp the Development Environment					
	2.1	Introduction to the MAF Development Environment	2-1				
	2.2	Configuring the Development Environment for Target Platforms	2-1				
	2.3	Configuring the Development Environment for Form Factors	2-3				
	2.4	Setting Up Development Tools for the iOS Platform	2-5				
	2.4.1	How to Install Xcode and iOS SDK	2-5				
	2.4.2	How to Set Up an iPhone or iPad	2-5				
	2.4.3	How to Set Up an iPhone or iPad Simulator	2-5				
	2.5	Setting Up Development Tools for the Android Platform	2-6				
	2.5.1	How to Install the Android SDK	2-6				
	2.5.2	How to Set Up an Android-Powered Device	2-7				
	2.5.3	How to Set Up an Android Emulator	2-7				
	2.5.3.1	Configuring the Android Emulator	2-8				
	2.5.3.2	Speeding Up the Android Emulator	2-10				
	26	Testing the Environment Setup	2-13				

3.1	Migrating an Application to MAF 2.1.0	3-1
3.2	Migrating to JDK 8 in MAF 2.1.0	3-1
3.3	Migrating Cordova Plugins from Earlier Releases to MAF 2.1.0	3-2
3.4	Migrating ADF Mobile Applications	3-3

3.4.1	What Happens When You Migrate an ADF Mobile Application	3-4
3.4.1.1	About Migrating Web Service Policy Definitions	3-6
3.4.2	What You May Need to Know About FARs in Migrated Applications	3-6
3.5	Migrating to New cacerts File for SSL in MAF 2.1.0	3-6

Preface

Welcome to Installing Oracle Mobile Application Framework.

Audience

This manual is intended for developers who want to install the Oracle Mobile Application Framework for use with Oracle JDeveloper to create mobile applications that run natively on devices.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see *Developing Mobile Applications with Oracle Mobile Application Framework*.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements (for example, menus and menu items, buttons, tabs, dialog controls), including options that you select.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates language and syntax elements, directory and file names, URLs, text that appears on the screen, or text that you enter.

1

Installing Mobile Application Framework with JDeveloper

This chapter describes how to install JDeveloper and the Mobile Application Framework (MAF) extension for application development.

This chapter includes the following sections:

- Introduction to Installing the MAF Extension with JDeveloper
- Installation Requirements for MAF Applications to be Deployed to the iOS Platform
- Installation Requirements for MAF Applications to be Deployed to the Android Platform
- Setting Up JDeveloper
- Installing the MAF Extension in JDeveloper

1.1 Introduction to Installing the MAF Extension with JDeveloper

Before you can create a MAF application using the MAF extension in JDeveloper, you need to:

 Ensure that you have any third-party software required to develop applications for the platform on which you intend to deploy your MAF application.

Note: You can deploy the same MAF application to all supported platforms without changing your application's code. You need the third-party software to test, debug, and deploy the MAF application on the target platform.

- Install JDeveloper using JDK 1.7
- Install the MAF extension in JDeveloper and specify JDK 1.8 in the dialog that appears after JDeveloper restarts following installation of the extension so that MAF applications compile using JDK 1.8

Following installation of the MAF extension in JDeveloper, you need to configure additional development tools for the platforms where you intend to deploy your MAF application. For more information, see Chapter 2, "Setting Up the Development Environment."

1.2 Installation Requirements for MAF Applications to be Deployed to the iOS Platform

Before you start creating a MAF application that you are planning to deploy to the iOS platform, ensure that you have the following available:

- A computer running Apple Mac OS X Version 10.9.5 or later
- Oracle JDeveloper (see Section 1.4, "Setting Up JDeveloper").
- Oracle JDeveloper extension for MAF (see Section 1.5, "Installing the MAF Extension in JDeveloper")
- Xcode and iOS SDK (see Section 2.4.1, "How to Install Xcode and iOS SDK")
- The most recent version of JDK1.8
- The most recent version of JDK1.7

Before you start deploying your application to a development environment (see the "Getting Started with Mobile Application Development" chapter in *Developing Mobile Applications with Oracle Mobile Application Framework*), decide whether you would like to use a mobile device or its simulator: if you are to use a simulator, see Section 2.4.3, "How to Set Up an iPhone or iPad Simulator"; if your goal is to deploy to a mobile device, ensure that, in addition to the components included in the preceding list, you have the following available:

- Various login credentials. For more information, see the "Deploying Mobile Applications" chapter in *Developing Mobile Applications with Oracle Mobile Application Framework*.
- iOS-powered device. For more information, see Section 2.4.2, "How to Set Up an iPhone or iPad."

1.3 Installation Requirements for MAF Applications to be Deployed to the Android Platform

Before you start creating a MAF application that you are planning to deploy to the Android, ensure that you have the following available:

- A computer running one of the following operating systems:
 - Microsoft Windows Vista
 - Microsoft Windows 7
 - Mac OS X
- The most recent version of JDK1.8
- The most recent version of JDK1.7
- Android SDK Manager (see Section 2.5, "Setting Up Development Tools for the Android Platform")
- Oracle JDeveloper (see Section 1.4, "Setting Up JDeveloper")
- Oracle JDeveloper extension for MAF (see Section 1.5, "Installing the MAF Extension in JDeveloper")

Before you start deploying your application to a development environment (see the "Getting Started with Mobile Application Development" chapter in *Developing Mobile Applications with Oracle Mobile Application Framework*), decide whether you would like to use a mobile device or its emulator: if you are to use an emulator, see Section 2.4,

"Setting Up Development Tools for the iOS Platform"; if your goal is to deploy to a mobile device, ensure that, in addition to the components included in the preceding list, you have the following available:

- Various login credentials. For more information, see the "Deploying Mobile Applications" chapter in *Developing Mobile Applications with Oracle Mobile Application Framework*.
- Android-powered device. For more information, see Section 2.5, "Setting Up Development Tools for the Android Platform."

1.4 Setting Up JDeveloper

Oracle JDeveloper and its MAF extension are essential tools used in developing MAF applications.

Before you begin:

Download and install the latest version of JDK 1.7.

This version of JDK is required by JDeveloper.

Download and install the latest version of JDK 1.8.

This version of JDK is required by the MAF extension.

 Download the 12.1.3.0.0 release of JDeveloper (Studio Edition) available at http://www.oracle.com/technetwork/developer-tools/jdev/downloads/index. html.

To install JDeveloper on a computer running the Windows platform:

- 1. In your file system, navigate to the directory that contains the JDeveloper executable file, then right-click that folder and select CMD Prompt Here As Administrator.
- **2.** Run the following command to explicitly install JDeveloper using the required JDK 1.7:

<fully_qualified_path_to_JDK7>\bin\java -jar <JDEV_12.1.3_jar>

For more information, see Oracle Fusion Middleware Installation Guide for Oracle JDeveloper.

To install JDeveloper on a computer running the Mac OS X platform:

- **1.** Open a Terminal window.
- **2.** Set the JAVA_HOME to Java 1.7 by running the following command:

export JAVA_HOME=\$(/usr/libexec/java_home -v1.7)

3. Verify that Java 1.7 is used by running the following command:

java -version

4. Using the same Terminal window, install JDeveloper by executing the following:

java -jar <JDEV_12.1.3_jar>

For more information, see the section about using Oracle JDeveloper on the Mac OS X platform in *Oracle Fusion Middleware Installation Guide for Oracle JDeveloper*.

To verify the installation of JDeveloper:

- 1. Check the *<JDEV_HOME>*\jdev\bin\jdev.conf file and confirm that the SetJavaHome property points to JDK 1.7.
- **2.** Start JDeveloper and select the Studio Developer (All Features) role when prompted.
- **3.** From the main menu, select Help > About > Version and ensure that the Java platform 1.7 is used, as Figure 1–1 shows.

Figure 1–1 Verifying JDK Version

About Oracle JDevelo	per 12c	
About Version Pro	nerties Extensions	<u>E</u> xport ▼
Q		
▲ Component	Version	~
Java(TM) Platform	1.7.0_25	
Oracle IDE	12.1.3.0.41.140521.1008	

1.5 Installing the MAF Extension in JDeveloper

You download the MAF extension using the Check for Updates menu in JDeveloper.

Once you have installed the MAF extension, you need to configure additional development tools for the platforms where you intend to deploy your MAF application. For more information, see Chapter 2, "Setting Up the Development Environment."

To download and install the MAF extension:

1. In JDeveloper, choose Help > Check for Updates.

Note: You might need to configure proxy settings on your development computer: on Windows, select **Tools > Preferences** from the main menu, and then **Web Browser and Proxy** from the tree on the left of the **Preferences** dialog; on Mac OS X, this option is accessed from **JDeveloper > Preferences**.

2. In the Select update source page that Figure 1–2 shows, select Official Oracle Extensions and Updates under the Search Update Centers, and then click Next.

😚 Check for Updates - Step 1 of 4			×
Select update source			R.
Source Updates Download Summary	Search for updates published to Update Centers, Automatically check for updates at startup Search Update Centers: Oracle Fusion Middleware Product http://www.orade.com/ocom/groups/p Official Oracle Extensions and Upd http://apex.oracle.com/pls/apex/f2p =u Open Source and Partners Extens http://www.orade.com/groups/p Internal Automatic Updates (12.1 http://de.us.oracle.com/center3.xml	or install an update from a bundle you have alread Proxy Settings ts ublic/@otn/documents/webcontent/156082.xml dates updatecenter:uc ions ublic/@otn/documents/webcontent/130355.xml 1.3+)	y downloaded.
	O Install From Local File Eile Name:		Browse

Figure 1–2 Checking for Updates in JDeveloper

Alternatively, if network access is not available, you can select the **Install From Local File** option. In this case, you need to point to the MAF extension file that you already downloaded to a directory on your development computer.

- **3.** In the **Select updates to install** dialog, select the **Mobile Application Framework** update.
- **4.** In the **License Agreements** page, shown in Figure 1–3, review *The Oracle Technology Network License Terms for Oracle Mobile*.

Note: You must comply with all of the license terms and conditions with respect to the Oracle Mobile Application Framework Program available at

http://www.oracle.com/technetwork/indexes/downloads/index.ht
ml.

5. Click I Agree.

🍘 Check for Updates - Step 2 o	yf 3 X
License Agreements	
Source	Before installing these updates, you must read and agree to the terms of their licenses. For each agreement, review the license text and click the I Agree button to indicate that you accept the terms.
License Agreements	License for Mobile Application Framework 2.1.0.0.41.141118.0913:
O Summary	Oracle Technology NetworkDeveloper License Terms
	Export Controls Export Laws and regulations of the United States and any other relevant localexport laws and regulations apply to the Programs. You agree that such export controllaws govern your use of the Programs (including technical data) and anyservices deliverables provided under this agreement, and you agree to complywith all such export laws and regulations (including "deemed export" and "deemed re-export" regulations). You agree that no data, information, program and/or materials resulting from services (or direct product thereof/will be exported, directly or indirectly, in violation of these laws, or willbe used for any purpose prohibited by these laws including, without limitation, nuclear, chemical, or biological weapons proliferation, or development ofmissile technology.
	Accordingly, you confirm:
	 You will not download, provide, make available or otherwise export orre-export the Programs, directly or indirectly, to countries prohibited byapplicable laws and regulations nor to citizens, nationals or residents offhose countries. You are not listed on the United States Department of Treasury lists ofSpecially Designated Nationals and Blocked Persons, Specially Designated Parorists, and Specially Designated Narcotic Traffickers, nor are you listedon the United States Department of Commerce Table of Denial Orders. You will not download or otherwise export or re-export the Programs, directlyor indirectly, to persons on the above mentioned lists. You will not use the Programs for, and will not allow the Programs to be used for, any purposesprohibited by applicable law, including, without limitation, for thedevelopment, design, manufacture or production of nuclear, chemical or biologicalweapons of mass destruction.
	Oracle Employees: Under no circumstances are OracleEmployees authorized to download software for the purpose of distributing it tocustomers. Oracle products are available to employees for internal use ordemonstration purposes only. In keeping with Oracle's trade complianceobligations under U
	I <u>Ag</u> ree
Help	< Back Next > Einish Cancel

Figure 1–3 Licensing Agreements for Mobile Application Framework Program

- 6. Click Next, and then click Finish.
- 7. Restart JDeveloper.
- **8.** Use the **Create JDK 8 Profile** dialog that Figure 1–4 shows to specify the path to the directory on your computer that contains JDK 1.8.

Figure 1–4 Creating JDK 8 Profile



Note: If you specify an invalid directory or directory that does not contain JDK 1.8, an error dialog is displayed.

You do not have to complete the Create JDK 8 Profile dialog the next time you use JDeveloper, unless you reinstall the MAF extension and choose not to preserve JDeveloper's system preferences.

- 9. Check whether or not MAF has been successfully added to JDeveloper:
 - Select File > New > From Gallery from the main menu to open the New Gallery dialog.

 In the Categories tree on the left, expand the Client Tier node and make sure it contains Mobile Application Framework (see Figure 1–5).

😚 New Gallery		×
Q		
<u>C</u> ategories:	Items:	Show All Descriptions
⊕-General ⊕-Business Tier ⊡-Client Tier ⊶ADF Desktop Integration Extension Development	MAF AMX Page Launches the Create MAF AMX Page dialog, in which Application Framework AMX Page (.amx) file. To en project or a file within a project in the Application N	h you create a new Mobile able this option, you must select a avigator.
Mobile Application Framework	MAF AMX Page Fragment	
Swing/AWT	🍓 MAF Feature	
⊕Web Tier	MAF Task Flow	
·Au Items		
Help		OK Cancel

Figure 1–5 Verifying MAF Installation

In addition, verify that you installed the correct version of MAF. To do so, select **Help > About** from the main menu, then select the **Extensions** tab on the About Oracle JDeveloper dialog, and then examine the extension list entries by searching for **Mobile Application Framework**, as Figure 1–6 shows.

Figure 1–6 Verifying MAF Version

About Version Properties Extensions			Entra
Q mobile application			
Name	Identifier	Version	Status
1obile Application Framework	oracle.maf	2.1.0.0.41.141118.0913	Fully Loaded
Nobile Application Framework Help	oracle.maf.doc	2.1.0.0.41.141118.0913	Fully Loaded
Nobile Application Framework Page Flow Design Tin	ne oracle.adf.pageflow.mobile.dt	2.1.0.0.41.141118.0913	Fully Loaded

In addition to the preceding steps, your development environment must be configured for target platforms and form factors. For more information, see Chapter 2, "Setting Up the Development Environment."

Setting Up the Development Environment

This chapter provides information on setting up and configuring the MAF environment for application development and deployment.

This chapter includes the following sections:

- Introduction to the MAF Development Environment
- Configuring the Development Environment for Target Platforms
- Configuring the Development Environment for Form Factors
- Setting Up Development Tools for the iOS Platform
- Setting Up Development Tools for the Android Platform
- Testing the Environment Setup

2.1 Introduction to the MAF Development Environment

After you install the MAF extension, as described in Chapter 1, "Installing Mobile Application Framework with JDeveloper," you may need to configure the development environment for the platforms where you intend to deploy your MAF application and configure form factors if you intend to test or deploy on a particular device. You may also need to install and configure additional third-party tools that allow you to package and deploy your MAF application on supported platforms.

For complete list of supported versions of development and runtime tools, see Oracle Mobile Application Framework Certification Matrix by following the Certification Information link on the MAF documentation page at http://www.oracle.com/technetwork/developer-tools/maf/documentation/.

2.2 Configuring the Development Environment for Target Platforms

For successful packaging and deployment of your application to platforms supported by MAF, JDeveloper must be provided with such information as the name of the platform and directories on your development computer that are to house the platform-specific tools and data. For convenience, MAF prepopulates JDeveloper Preferences with these settings. Depending on several factors related to the application signing, you may need to edit some of the fields.

Before you begin:

Download and install JDeveloper and the MAF extension, as described in Chapter 1, "Installing Mobile Application Framework with JDeveloper." Depending on your target platform, download and configure either the Android SDK (see Section 2.5.1, "How to Install the Android SDK") or iOS SDK and Xcode (see Section 2.4.1, "How to Install Xcode and iOS SDK").

To configure your environment for target platforms:

- 1. Select Tools > Preferences from JDeveloper's main menu to open Preferences.
- In the Preferences dialog, select either Mobile Application Framework > Android Platform or Mobile Application Framework > iOS Platform from the tree to open a page that contains the path and configuration parameters for the supported platforms, as Figure 2–1 and Figure 2–2 show.

Each platform-specific page hosts the preferences for the platform SDK (Android or iOS), collecting any necessary information such as the path that MAF needs to compile and deploy either Android or iOS projects:

- For the Android platform, specify the following:
 - The Android SDK location on your computer.
 - The local directory of your target Android platform.
 - The Android build tools location on your computer.
 - Information on the signing credentials.

Figure 2–1 Configuring Platform Preferences for Android



- For the iOS platform, specify the following:
 - Location of the iTunes media files, including the mobile applications that are synchronized to the iOS-powered device.
 - The iOS-powered device signing information (see the "Setting the Device Signing Options" section in *Developing Mobile Applications with Oracle Mobile Application Framework*).

🕐 Preferences		×
Q Search	Mobile Application Framework: iOS Platform	
CSS Editor	"Automatically Add to [Tunes" Directory:	
🖻 Database	\Users\My_User_Name\Music\iTunes\iTunes Media\Automatically Add to iTunes	
Data Controls Panel	Example: /Users/My_User_Name/Music/iTunes/iTunes Media/Automatically Add to iTunes	~
主 Debugger	- Device Signing	
🕀 Deployment	These fields are explicitly field with the second action for the extent IOC device.	
🗄 Diagrams	These fields are required if deploying to, or packaging for, an actual IOS device.	
External Editor	Provisionina Profile:	
File Templates		
- File Types	Signing Identity:	
Global Ignore List		
🕀 Http Analyzer		
Issues		
JavaScript Editor		
🖅 JSP and HTML Visual Editor		
🕀 Maven 🔪		
Merge		
Mobile Application Framewo		
Android Platform		
Containerization		
iOS Platform		
Mouseover Popups		
News		
Help	OK Cancel	

Figure 2–2 Configuring Platform Preferences for iOS

2.3 Configuring the Development Environment for Form Factors

A form factor is a specific device configuration. Each form factor is identified by a name that you specify for it and contains information on the specified resolution denoted by pixel width and pixel height.

Since form factors defined in preferences are used in the MAF AMX page Preview tab (see the "Using the Preview" section in *Developing Mobile Applications with Oracle Mobile Application Framework*), you may choose to perform this configuration if you are planning to include a MAF AMX application feature as part of your MAF application and you do not want to accept the default settings. During development, you can select or switch between various form factors to see how a MAF AMX page is rendered. You can also see multiple form factors applied to the same page using the split screen view.

For more information, see the "About the maf-config.xml File" section in *Developing Mobile Applications with Oracle Mobile Application Framework*.

Before you begin:

Download and install JDeveloper and the MAF extension, as described in Chapter 1, "Installing Mobile Application Framework with JDeveloper."

To configure the form factors:

- Open Preferences by selecting Tools > Preferences from the main menu in JDeveloper.
- **2.** In the **Preferences** dialog that Figure 2–3 shows, select **Mobile Application Framework** from the tree on the left.

Preferences								
Q, Search	Mobile	Application Framew	ork					
Diagrams	Available F	Form Factors				4	<mark>⊨ X</mark> Ma	ore Actions
External Editor	Default	▲ Name*	Platform*	Model	Default Orientation	Width*	Height*	Scale F
File Templates		Android Extra Large	Android		Landscape	1280	800	2
	0	Android High	Android		Portrait	480	800	1
B Http Apalyzer		Android Low	Android		Portrait	240	400	1
- Issues	Ĭ	Android Medium	Android		Portrait	320	480	1
JavaScript Editor	ŏ	iPad Non-Retina	iOS	iPad	Landscape	1024	768	1
JSP and HTML Visual Editor	ĬĬŏ	iPad Retina	iOS	iPad	Landscape	2048	1536	- 2
Maven	ĬĬŏ	iPhone Retina (3,5 inch)	iOS	iPhone	Portrait	640	960	- 2
Merge	ĬĬŏ	iPhone Retina (4.0 inch)	ios	iPhone	Portrait	640	1136	2
Mobile Application Framework	ĬĬŏ	iPhone Retina (4.7 inch)	ios	iPhone	Portrait	750	1334	2
Mouseover Popups		iPhone Retina (4.7 inch)	105	iPhone	Portrait	1090	1020	2
News		FIONE Reuna (5.5 men)	103	FILITIE	Foruait	1080	1920	5
Oracle Cloud								
Profiler								
Resource Bundle								
Shortcut Keye								
a Swing GLII Builder								
Task Tags								
TopLink								
∃ UML								
Usage Reporting								
Versionina 🗸 🗸								
Help						ОК		Cancel

Figure 2–3 Defining Form Factors

The **Mobile Application Framework** page is populated with available form factors and the default is set to Android Low.

This preference page allows you to create and manage a set of named form factors that combine a screen resolution size and platform.

- **3.** To create a new form factor, click the green plus sign (New), and then set the following:
 - **Name**: a meaningful string that is used to identify the form factor.
 - Platform: the platform of the mobile device.
 - Model: the type of the mobile device.
 - Default Orientation: the default device orientation used in the MAF AMX page Preview tab. It might be Portrait or Landscape. Select this setting from the drop-down list of values. The default value is Portrait and it is prepopulated during creation of the new form factor.
 - Width: width, in pixels. This value must be a positive integer, and its input is validated.
 - Height: height, in pixels. This value must be a positive integer, and its input is validated.
 - Scale Factor: the display scale factor. This value must be either one of 1.0, 2.0, or 3.0.

Note: If you do not set the name and resolution for your form, MAF will display an error message.

- 4. If you need to revert to default settings, click More Actions > Restore Defaults.
- 5. Click OK to finalize your settings.

2.4 Setting Up Development Tools for the iOS Platform

In addition to general-purpose tools listed in Section 1.1, "Introduction to Installing the MAF Extension with JDeveloper," you might want to set up an iPhone or iPad when getting ready for development of a MAF application for the iOS platform (see Section 2.4.2, "How to Set Up an iPhone or iPad").

Since iPhone and iPad simulators are included in the iOS SDK installation, which, in turn, is included in Xcode installation, you do not need to separately install them. For more information, see Section 2.4.3, "How to Set Up an iPhone or iPad Simulator."

2.4.1 How to Install Xcode and iOS SDK

You download Xcode from http://developer.apple.com/xcode/. This download includes the iOS SDK.

After installing Xcode, you have to run it at least once and complete the Apple licensing and setup dialogs. If these steps are not performed, any build and deploy cycle from JDeveloper to Xcode or device simulator will fail with a "Return code 69" error.

Note: Since older versions of Xcode and iOS SDK are not available from the Mac App Store, in order to download them you must obtain an Apple ID from http://appleid.apple.com, and then register this Apple ID with the Apple Developer Program to gain access to the Apple developer site at http://developer.apple.com.

2.4.2 How to Set Up an iPhone or iPad

In your MAF application development and deployment, you can use either the iPhone, iPad, or their simulators (see Section 2.4.3, "How to Set Up an iPhone or iPad Simulator"). If you are planning to use an actual iPhone or iPad, which is preferable for testing (see the "Testing MAF Applications" section in *Developing Mobile Applications with Oracle Mobile Application Framework*), you need to connect it to your computer to establish a link between the two devices.

To deploy to an iOS-powered device, you need to have an iOS-powered device with a valid license, certificates, and distribution profiles. For more information, see the "Deploying Mobile Applications" chapter in *Developing Mobile Applications with Oracle Mobile Application Framework*.

Note: Since Apple's licensing terms and conditions may change, ensure that you understand them, comply with them, and stay up to date with any changes.

2.4.3 How to Set Up an iPhone or iPad Simulator

In your MAF application development and deployment, you can use either the iOS-powered device itself (see Section 2.4.2, "How to Set Up an iPhone or iPad") or its simulator. Deploying to a simulator is usually much faster than deploying to a device, and it also means that you do not have to sign the application first.

A simulator can be invoked automatically, without any additional setup.

Note: Before attempting to deploy your application from JDeveloper to a device simulator, you must first run the simulator.

If you are planning to use web services in your application and you are behind a corporate firewall, you might need to configure the external network access. You do so by modifying the network settings in the System Preferences on your development computer. For more information, see the "Configuring the Browser Proxy Information" section in *Developing Mobile Applications with Oracle Mobile Application Framework*.

2.5 Setting Up Development Tools for the Android Platform

In addition to the general-purpose tools listed in Section 1.1, "Introduction to Installing the MAF Extension with JDeveloper," you might want to set up an Android-powered device when getting ready for development of a MAF application for the Android platform (see Section 2.5.2, "How to Set Up an Android-Powered Device").

Since emulators are included in the Android SDK installation, you do not need to separately install them. However, you cannot use an emulator until you create its configuration (see Section 2.5.3, "How to Set Up an Android Emulator").

To develop for the Android platform, you can use any operating system that is supported by both JDeveloper and Android.

For more information, see the "Developer Tools" section of the Android Developers website at http://developer.android.com/tools/index.html.

2.5.1 How to Install the Android SDK

Android SDK includes development tools that you need to build applications for Android-powered devices. Since the Android SDK is modular, it allows you to download components separately depending on your target Android platform and your application requirements.

When choosing the platform, keep in mind that MAF supports Android 4.0.3 or later.

Before you begin:

Ensure that your environment meets the operating system, JDK version, and hardware requirements listed in the "Get the Android SDK" section of the Android Developers website at http://developer.android.com/sdk/index.html.

Note: Ant and Linux requirements are not applicable to the MAF development environment; Eclipse might be applicable depending on your IDE of choice.

To install the Android SDK:

- Download the Android SDK starter package from http://developer.android.com/sdk/index.html.
- Complete the installation by following the instructions provided in the "Setting Up an Existing IDE" section of the Android Developers website at http://developer.android.com/sdk/installing.html.

Note: If you are not planning to use Eclipse, skip step 3 in the Android SDK installation instructions.

2.5.2 How to Set Up an Android-Powered Device

In your MAF application development and deployment, you can use either the Android device itself, which is preferable for testing (see the "Testing MAF Applications" section in *Developing Mobile Applications with Oracle Mobile Application Framework*), or an emulator (see Section 2.5.3, "How to Set Up an Android Emulator.").

For information on how to set up the Android-powered device, follow the instructions from the "Using Hardware Devices" section of the Android Developers website at http://developer.android.com/tools/device.html.

Note: You might experience issues when using USB connectivity for the device-based debugging. For more information, see the "Testing and Debugging MAF Applications" chapter in *Developing Mobile Applications with Oracle Mobile Application Framework*.

Your target Android-powered device might not be listed in the USB device driver's.inf file, resulting in the failure to install the Android Debug Bridge (ADB). You can eliminate this issue as follows:

- 1. Find the correct values for your device.
- Update the [Google.NXx86] and [Google.NTamd64] sections of the android_ winusb.inf file.

For more information, see the "Google USB Driver" section of the Android Developers website at http://developer.android.com/sdk/win-usb.html.

2.5.3 How to Set Up an Android Emulator

In your MAF application development and deployment, you can use either the Android device itself (see Section 2.5.2, "How to Set Up an Android-Powered Device") or its emulator. Deploying to an emulator is usually much faster than deploying to a device, and it also means that you do not have to sign the application first.

For information on how to create an emulator configuration called Android Virtual Device (AVD), follow the instructions from the "Managing Virtual Devices" section of the Android Developers website at

http://developer.android.com/tools/devices/index.html. When creating an AVD through the Create New Android Virtual Device dialog (see "Managing AVDs with AVD Manager" at

http://developer.android.com/tools/devices/managing-avds.html), review all the settings to ensure that configuration matches what you are planning to emulate. In particular, you should verify the following:

- The Target field should define the desired Android platform level for proper emulation.
- The CPU/ABI field should reflect the ARM or Intel Atom system image (see Section 2.5.3.2.1, "Configuring AVD for Intel HAXM").
- The SD card field should be defined based on whether the application uploads files or files install themselves to the SD card.

Default settings for the Hardware field (see the "Hardware Options" table at http://developer.android.com/tools/devices/managing-avds.html#hardwareo pts) should be acceptable for a typical MAF application. For additional hardware capabilities you may want to use in your application, such as cameras or geolocation services, create new properties.

You need to create an AVD for each Android platform on which you are planning to test your application.

For information on how to use the emulator, see the "Using the Android Emulator" section in the Android Developers website at

http://developer.android.com/tools/devices/emulator.html.

2.5.3.1 Configuring the Android Emulator

After the basic Android emulator setup is complete, you may choose to perform the following configurations:

- Save the emulator state (see Section 2.5.3.1.1, "Saving the Emulator State")
- Create, save, and reuse the SD card (see Section 2.5.3.1.2, "Creating, Saving, and Reusing the SD Card")
- Configure the network (see Section 2.5.3.1.3, "Configuring the Network")
- Configure the network proxy (see Section 2.5.3.1.4, "Configuring the Network Proxy")

2.5.3.1.1 Saving the Emulator State You can reduce the emulator's load time by saving the emulator state or reusing the saved state. To do so, you manipulate the avd files or folders that are located in the C:\Users\username\.android\avd directory (on a Windows computer). Each avd folder contains several files, such as userdata.img, userdata.qemu.img, and cache.img. You can copy the cache.img file to another emulator's avd folder to use that state with another emulator.

Alternatively, you can use the command line to run relevant commands, such as, for example, -snapshot-list, -no-snapstorage, and so on. You can access these commands through emulator -help command.

Caution: When using this utility, keep in mind that in the process of loading, all contents of the system, including the user data and SD card images, will be overwritten with the contents they held when the snapshot was made. Unless saved in a different snapshot, any changes will be lost.

2.5.3.1.2 Creating, Saving, and Reusing the SD Card The "SD Card Emulation" section of the Android Developers website at

http://developer.android.com/tools/devices/emulator.html#sdcard lists reasons for creating, saving, and reusing the SD card. You can perform these operations by executing the following commands:

To create an SD card:

C:\android sdk directory\tools>mksdcard -1 SD500M 500M C:\Android\sd500m.img

• To list existing AVDs:

C:\android sdk directory\tools>android list avd

This produces a listing similar to the following:

```
Name: AndroidEmulator1
Device: Nexus S (Google)
Path: C:\Users\username\.android\avd\AndroidEmulator1.avd
Target: Android 4.2.2 (API level 17)
Tag/ABI: default/x86
Skin: 480x800
------
Name: AndroidEmulator2
Device: Nexus S (Google)
Path: C:\Users\username\.android\avd\AndroidEmulator2.avd
Target: Android 4.2.2 (API level 17)
Tag/ABI: default/armeabi-v7a
Skin: 480x800
Sdcard: 500M
```

• To start the AndroidEmulator2 with the SD card that has just been created:

C:\Android\android sdk directory\tools>emulator -avd AndroidEmulator2 -sdcard C:\Android\sd500m.img

To list the running Android emulator instances:

C:\Android\android sdk directory\platform-tools>adb devices

To copy a test image to the SD card (this requires the emulator to restart):

C:\Android\sdk\platform-tools>adb push test.png sdcard/Pictures 85 KB/s (1494 bytes in 0.017s)

For more information, see the Android Tools Help at http://developer.android.com/tools/help/index.html.

2.5.3.1.3 Configuring the Network From the Android emulator, you can access your host computer through the 10.0.2.2 IP. To connect to the emulator from the host computer, you have to execute the adb command from a command line on your development computer or from a script to set up the port forwarding.

To forward socket connections, execute

adb forward local remote

using the following forward specifications:

- tcp:port
- localabstract:unix domain socket name
- localreserved:unix domain socket name
- localfilesystem:unix domain socket name
- dev:character device name
- jdwp:process pid (remote only)

For example, an arbitrary client can request connection to a server running on the emulator at port 55000 as follows:

```
adb -e forward tcp:8555 tcp:55000
```

In this example, from the host computer, the client would connect to localhost:8555 and communicate through that socket.

For more information, see the "Android Debug Bridge" section in the Android Developers website at http://developer.android.com/tools/help/adb.html.

2.5.3.1.4 Configuring the Network Proxy If your development computer is behind a corporate firewall, you might need to configure a proxy by using one of the following techniques:

1. Execute this command to start the emulator and initiate its connection with the browser:

emulator -avd myavd -http-proxy myproxy

- 2. Start the emulator and then use its Settings utility as follows:
 - 1. Select Wireless & Networks
 - 2. Select Mobile Networks > Access Point Names
 - 3. Select the appropriate internet option
 - 4. Set the proxy, port, username, and password using the Edit access point list

2.5.3.2 Speeding Up the Android Emulator

The Intel Hardware Accelerated Execution Manager (Intel HAXM) is designed to accelerate the Android-powered device emulator by making use of Intel drivers.

The Intel HAXM is available for computers running Microsoft Windows, Mac OS X, and a separate kernel-based virtual machine option (KRM) for Linux. See http://software.intel.com/en-us/android/articles/intel-hardware-accelerated-execution-manager to access installation guides and detailed descriptions of system requirements for each operating system.

Regardless of which operating system your development computer is running on, it must have the following:

 Version 17 or later of the Android SDK installed (see Section 2.5.1, "How to Install the Android SDK").

Note: Currently, the recommended version for MAF development is 21.

- Intel processor with support for Intel VT-x, EM64T and Execute Disable (XD) Bit functionality at the BIOS level.
- At least 1 GB of available RAM.

To download the Intel HAXM, either use the Android SDK Manager (see *Speeding Up the Android Emulator on Intel Architecture*) or use the following Intel locations:

- Download for Microsoft Windows
- Download for Mac OS X
- Download for Linux

To install the Intel HAXM, follow the steps described in the "Speeding Up the Android Emulator on Intel Architecture" article available at

http://software.intel.com/en-us/android/articles/speeding-up-the-android-e mulator-on-intel-architecture. It is particularly important to configure AVD (see Section 2.5.3.2.1, "Configuring AVD for Intel HAXM").

If your development computer is running either Microsoft Windows 8.*n* or later, or Mac OS X 10.9.*n* or later, you have to apply a Hotfix provided by Intel before using the emulator with the Intel HAXM.

Note: If you do not apply the Hotfix, your computer will freeze and you will lose your work.

To download the Hotfix, use the following locations:

- Download for Microsoft Windows
- Download for Mac OS X

For more information, see the following:

- Installation Guide and System Requirements Windows
- Installation Guide and System Requirements Mac OS X
- Installation Guide and System Requirements Linux

2.5.3.2.1 Configuring AVD for Intel HAXM When enabling the Intel HAXM, ensure that you download the Intel system image for the Android API level using the Android SDK Manager (see Figure 2–4). The following steps described in *Speeding Up the Android Emulator on Intel Architecture* guide you through the configuration process:

- After you have installed the Android SDK, open the SDK Manager and then find the Intel HAXM in the extras section.
- Select Intel x86 Emulator Accelerator (HAXM) and click Install packages.

Once you have installed the package, the status changes to Installed, which is not accurate: the SDK only copies the Intel HAXM executable on your computer; you have to manually install the executable.

Figure 2–4	Downloading Intel S	ystem Image in Andr	oid SDK Manager
•			

- N:	ame	ΔΡΙ	Rev	Status
	(Art 5)	All		Status
	Extras			
	Android Support Repository		4	🖊 Not installed
	🛃 Android Support Library		12	🖊 Update available: rev. 19
	🛃 Google AdMob Ads SDK		11	🖊 Not installed
	🛃 Google Analytics App Tracking SDK		3	🖊 Not installed
	🛃 [Deprecated] Google Cloud Messaging for An		3	installed 🥭
	🛃 Google Play services for Froyo		12	🖊 Not installed
	🛃 Google Play services		14	🖊 Not installed
	🛃 Google Repository		5	🖊 Not installed
	🛃 Google Play APK Expansion Library		3	🖊 Not installed
	🛃 Google Play Billing Library		5	🖊 Not installed
	🛃 Google Play Licensing Library		2	🖊 Not installed
	🛃 Google USB Driver		8	& Not compatible with Mac C
	🛃 Google Web Driver		2	🖊 Not installed
	🕢 Intel x86 Emulator Accelerator (HAXM)		3	📇 Installed
how:	Updates/New Installed Obsolete Select	New or U	pdates	Install 11 packages
ort by:	API level Repository Desele	ct All		Delete 3 packages

- To install the Intel HAXM executable, depending on your development platform search your hard drive for one of the following:
 - On Windows, search for IntelHaxm.exe

- On Mac OS X, search for IntelHaxm.dmg

If you accepted default settings, the executable should be located at C:\Program Files\Android\android-sdk\extras\Intel\Hardware_Accelerated_Execution_ Manager\IntelHaxm.exe on Windows.

The Intel HAXM only functions in combination with one of the Intel Atom processor x86 system images, which are available for Android 2.3.3 (API 10), 4.0.3 (API 15), 4.1.2 (API 16), 4.2.2 (API 17), 4.4 (API 19), 4.4W (API 20), 5.0 (API 21). These system images can be installed exactly like the ARM-based images through the Android SDK Manager.

Figure 2–5 Installing Intel Atom System Image

ickages		4.01		Status I
Na 🖷 Na	me	API	Kev.	Status
	Coople APIc	19	2	Not installed
	Google APIS	19	2	Not installed
Sources for Android SDK		19	2	Not installed
) - E	Android 4.3 (API 18)			
	SDK Platform	17	2	Installed
	A Samples for SDK	17	1	Not installed
ĥ	ARM FABL v7a System Image	17	2	
/	Intel x86 Atom System Image	17	1	Not installed
1	MIPS System Image	17	1	Not installed
j	Google APIs	17	3	Not installed
Sources for Android SDK		17	1	Not installed
) 🕨 🔛	Android 4.1.2 (API 16)			•
) 🕨 🔛	Android 4.0.3 (API 15)			
🕨 🕨 🔛	Android 4.0 (API 14)			
	A 1 110 0 (ADL10)			
how:	🗹 Updates/New 🗹 Installed 🛛 Obs	olete Select New or L	<u>Jpdates</u>	Install 5 packages
ort by:	API level Repository	Deselect All		Delete 3 packages

To complete the process, use the AVD Manager to create a new virtual device that has hardware-accelerated emulation by selecting **Intel Atom (x86)** as the CPU/ABI, (see Figure 2–6).

Note: This option appears in the list only if you have the Intel x86 system image installed.

AVD Name:	IntelAVD
Device:	Nexus S (4.0", 480 × 800: hdpi)
Target:	Android 4.2.2 - API Level 17
CPU/ABI:	Intel Atom (x86)
Keyboard:	🗹 Hardware keyboard present
Skin:	✓ Display a skin with hardware controls
Front Camera:	None
Back Camera:	None
Memory Options:	RAM: 343 VM Heap: 32
Internal Storage:	200 MiB ‡
Internal Storage: SD Card:	200 MiB ÷
Internal Storage: SD Card:	200 MiB ÷ ● Size: MiB ÷
Internal Storage: SD Card:	200 MiB ÷ • Size: MiB ÷ • File: Browse
Internal Storage: SD Card: Emulation Options:	Size: MiB Size: File: Browse Snapshot Use Host GPU
Internal Storage: SD Card: Emulation Options:	200 MiB ÷ • Size: MiB ÷ • File: Browse • Snapshot Use Host GPU
Internal Storage: SD Card: Emulation Options:	200 MiB ÷ • Size: MiB ÷ • File: Browse Snapshot Use Host GPU ting AVD with the same name
Internal Storage: SD Card: Emulation Options:	200 MiB ÷ • Size: MiB ÷ • File: Browse • Snapshot Use Host GPU ting AVD with the same name

Figure 2–6 Creating Accelerated AVD

2.6 Testing the Environment Setup

You can test your environment setup as follows:

- 1. In JDeveloper, open the HelloWorld sample application by selecting the HelloWorld.jws file (see the "Mobile Application Framework Sample Applications" appendix in *Developing Mobile Applications with Oracle Mobile Application Framework*).
- **2.** Select Application > Deploy from the main menu.

For more information, see the "Deploying Mobile Applications" chapter in *Developing Mobile Applications with Oracle Mobile Application Framework*.

- **3.** From the drop-down menu, select the deployment profile for the platform to which you wish to deploy the application.
- **4.** Since using an iOS-powered device simulator or Android-powered device emulator to test the environment setup is preferable because it does not require signing of the application, you should select one of the following deployment actions using the **Deploy** dialog:
 - For iOS, select **Deploy application to simulator**, as Figure 2–7 shows.

Deploy 105_MOBILE_NA Deployment Action	TTVE_archive1	×
Deployment Action	Select a deployment action from the list below. Deploy application to simulator Deploy to distribution package Deploy to iTunes for synchronization to device Deploy to iTunes for synchronization to device Deploy to iTunes for synchronization to device	
Help	< Back Next > Einish Cancel	

Figure 2–7 Selecting Deployment Action for iOS

• For Android, select **Deploy application to emulator**, as Figure 2–8 shows. Ensure that the emulator is running before you start the deployment.

Figure 2–8 Selecting Deployment Action for Android

Deploy ANDROID_MOBI	LE_NATIVE_archive1	×
Deployment Action		
Deployment Action Summary	Select a deployment action from the list below. Deploy application to device Deploy application to emulator Deploy application to package Deploy the mobile application to an Android device connected to the development machine	
	עריין איז	
Help	< Back Next > Finish Cancel	

5. Click **Next** on the Deploy dialog to verify the Summary page, and then click **Finish**.

For more information, see one of the following sections in *Developing Mobile Applications with Oracle Mobile Application Framework*:

"How to Deploy an iOS Application to an iOS Simulator"

"How to Deploy an Android Application to an Android Emulator"

For more information on deployment, see the "Deploying Mobile Applications" chapter in *Developing Mobile Applications with Oracle Mobile Application Framework*

After a successful deployment (which might take a few minutes), your iOS-powered device simulator or Android-powered device emulator will display the HelloWorld application icon that you have to activate to launch the application.

Migrating Your Application to MAF 2.1.0

This chapter provides information about migrating applications created using earlier releases of MAF and ADF Mobile to MAF 2.1.0.

This chapter includes the following sections:

- Migrating an Application to MAF 2.1.0
- Migrating to JDK 8 in MAF 2.1.0
- Migrating Cordova Plugins from Earlier Releases to MAF 2.1.0
- Migrating ADF Mobile Applications
- Migrating to New cacerts File for SSL in MAF 2.1.0

3.1 Migrating an Application to MAF 2.1.0

MAF 2.1.0 uses newer versions of Apache Cordova and Java. It also changes the way that JDeveloper registers plugins in your MAF application. For SSL, it delivers a cacerts file that contains new CA root certificates.

Read the subsequent sections in this chapter that describe how these changes impact the migration of your MAF application to MAF 2.1.0.

Finally, MAF 2.1.0 delivers an updated SQLite database and JDBC driver. Review, and migrate as necessary, any code in your migrated MAF application that connects to the SQLite database. For more information about how to connect to the SQLite database, see the "Using the Local SQLite Database" section in the *Developing Mobile Applications with Oracle Mobile Application Framework*.

3.2 Migrating to JDK 8 in MAF 2.1.0

MAF applications that you create in MAF 2.1.0 use JDK 8. You specify the location of your JDK 8 installation the first time you start JDeveloper after installing the MAF extension, as described in Section 1.5, "Installing the MAF Extension in JDeveloper."

If you migrate a MAF application that compiled with an earlier version of Java, note that MAF 2.1.0 requires JDK 8 and compiles applications using the Java SE Embedded 8 compact2 profile. When you open a migrated application in MAF 2.1.0 for the first time, JDeveloper makes the following changes:

 Renames the configuration file that specifies the startup parameters of the JVM from cvm.properties to maf.properties. For more information about the maf.properties file, see the "How to Enable Debugging of Java Code and JavaScript" section in *Developing Mobile Applications with Oracle Mobile Application Framework*. Replaces instances (if any) of the following import statement in the application's Java source files:

com.sun.util.logging

With:

java.util.logging

Replaces the following entries in the application's logging.properties file

```
.handlers=com.sun.util.logging.ConsoleHandler
.formatter=com.sun.util.logging.SimpleFormatter
```

With:

```
.handlers=java.util.logging.ConsoleHandler
.formatter=java.util.logging.SimpleFormatter
```

For more information about the logging.properties file, see the "How to Configure Logging Using the Properties File" section in *Developing Mobile Applications with Oracle Mobile Application Framework*.

3.3 Migrating Cordova Plugins from Earlier Releases to MAF 2.1.0

MAF applications developed using earlier releases of MAF registered plugins in the maf-application file. This release of MAF registers plugins in the maf-plugins.xml file. JDeveloper makes the following changes to an application from an earlier release that uses plugins when you migrate the application:

Comments out entries in the maf-application.xml file that referenced plugins.
 For example, JDeveloper comments out entries such as the following:

```
<!--<adfmf:cordovaPlugins>
<adfmf:plugin fullyQualifiedName="BarcodeScanner"
implementationClass="com.phonegap.plugins.
barcodescanner.BarcodeScanner" platform="Android"
name="BarcodeScanner">
.....
</adfmf:cordovaPlugins>-->
```

 Registers the plugin in the maf-plugins.xml file, as shown in the following example:

```
<cordova-plugins>
...
<cordova-plugin id="c3" pluginId="org.apache.cordova.barcodeScanner">
        <platform id="p3" name="ios" enabled="true"/>
        <platform id="p4" name="android" enabled="false"/>
        </cordova-plugin>
     </cordova-plugins>
```

To complete the migration and make sure that your migrated MAF application can use the plugins it used previously, verify that the:

Version of the plugin is supported by MAF.

MAF applications in 2.1.0 use Cordova 3.6.3 on Android and Cordova 3.7.0 on iOS.

Obtain a newer version of the plugin if the plugin was created using an earlier release of Cordova.

• Set the relative path to the plugin so that the MAF application's maf-plugins.xml file correctly references the plugin. For more information, see the "Registering Additional Plugins in Your MAF Application" section in *Developing Mobile Applications with Oracle Mobile Application Framework*.

If the maf-plugins.xml file does not correctly reference a plugin using a relative path, the overview editor for the maf-application.xml file's **Path*** field which requires a value is empty and the maf-plugins.xml displays a validation failure, as shown in Figure 3–1.

Figure 3–1 MAF Application that Does Not Specify Path to Plugin

maf-application.xr	nl ×				
					?
Application	Core Plugins				
Plugins					
Feature References	Additional Plugins				
Preferences	Register any additional plugins re	equired by your application. Adding a r	lugio will register it in maf-plugios, yml		
Security		equired by your application ridding a p			. .
	Plugins:			<u>]</u>	~ ×
	Plugin Id	Name	Path*	Version	
	BarcodeScanner				
	_prg.apache.cordova.barcodes	Scanner			
	Indicate which platforms to e	nable the plugin for:			
	Android				
	✓ iOS				
naf-plugins.xn	nl - Editor				
maf-plugins.xml	×.				
Q. Find	L ,				
xml ve</td <td>ersion="1.0" encoding="UTF-</td> <td>Required attribute 'path' missi</td> <td>ng</td> <td></td> <td>more</td>	ersion="1.0" encoding="UTF-	Required attribute 'path' missi	ng		more
□ <maf-plu< td=""><td>ugins xmlns:xsi="http://www</td><td>Add Attribute 'path'</td><td>-</td><td></td><td>indice </td></maf-plu<>	ugins xmlns:xsi="http://www	Add Attribute 'path'	-		indice
= <cord< td=""><td>pva-plugins></td><td>Suppress "Required Attribute Miss</td><td>ing" By IDeveloper Name (Suppress Pr</td><td>ocessing Instruction)</td><td></td></cord<>	pva-plugins>	Suppress "Required Attribute Miss	ing" By IDeveloper Name (Suppress Pr	ocessing Instruction)	
E <	rdova-plugin id="c2" plugir	Could not find the plugin yml fi	e for the Cordova plugin. Please	specify a directory containing plugin	- Maria
cplatform id="pl" name="and and plugin.xml higher the Cordova plugin. Please specify a directory containing plugin.xml in containing plugin.xml			more		
<1	platform id="p2" name="ios'	Suppress "Invalid Plugin Path" By	JDeveloper Name (Suppress Processing	(Instruction)	
		4			
	platform id="p3" name="ios"	<core-cordova-plugin id="</td"><td>"c1" pluginId="org.apache.c</td><td>ordova.camera"/></td><td></td></core-cordova-plugin>	"c1" pluginId="org.apache.c	ordova.camera"/>	
4	platform id="p4" name="and:	<cordova-plugin <="" id="c2" td=""><td>pluginId="BarcodeScanner"></td><td></td><td></td></cordova-plugin>	pluginId="BarcodeScanner">		
<td>ordova-plugin></td> <td><pre><platform <nlatform="" id="p2" name:="" name:<="" pre=""></platform></pre></td> <td>="android" enabled="true"/> ="ios" enabled="false"/></td> <td></td> <td></td>	ordova-plugin>	<pre><platform <nlatform="" id="p2" name:="" name:<="" pre=""></platform></pre>	="android" enabled="true"/> ="ios" enabled="false"/>		
<td>dova-plugins></td> <td><pre> vpraciorm ru= p2 name;</pre></td> <td>- 102 Ellapied- 19196 /></td> <td></td> <td></td>	dova-plugins>	<pre> vpraciorm ru= p2 name;</pre>	- 102 Ellapied- 19196 />		
<td>rugins></td> <td></td> <td></td> <td></td> <td></td>	rugins>				

3.4 Migrating ADF Mobile Applications

MAF automatically migrates the configuration of applications written in Versions 11.1.2.3.0 and 11.1.2.4.0 of ADF Mobile. After you open the workspace (.jws) file of an ADF Mobile application, MAF alerts you that the application is not the current version by presenting the Open Warning dialog (illustrated in Figure 3–2), that prompts you to continue with the migration, or dismiss the dialog and close the file.

Figure 3–2 Open Warning Dialog

Open Warn	ing 💽
1	You are about to migrate the application C:\JDeveloper\mywork\mobileApp1\mobileApp1.jws to JDeveloper version 12.1.3.0.0 file format. This operation will also migrate all projects contained in the application.
_	Once the application and its contents are migrated, you will not be able to open the application or its projects using an older release. You may want to back up the application contents before proceeding.
	Do you want to migrate these files?
Help	Yes No

MAF writes the status of the migration to the Log window, as illustrated by Figure 3–3. The migration process also logs the following warning if it detects that the application to migrate uses the old configuration service API.

The MAF 2.0 Configuration Service API is not backwards compatible with previous versions and cannot be migrated automatically. Refer to Section 9.3 "Migrating the Configuration Service API" in Oracle Fusion Middleware Developing Mobile Applications with Oracle Mobile Application Framework 2.0. for information on migrating to the new API.

For more information, see the "Migrating the Configuration Service" section in *Developing Mobile Applications with Oracle Mobile Application Framework*.

Figure 3–3 Migration Log

🔊 🔍
Messages - Log ×
Mar 12, 2014 12:50:25 PM oracle.security.jps.util.JpsUtil disableAudit
INFO: JpsUtil: isAuditDisabled set to true
[12:50:41 PM] Migration started.
Mar 12, 2014 12:50:42 PM oracle.adfdtinternal.model.adapter.webservice.utils.WSMPolicyUtil
migrateWSConnections
INFO: WSMPolicyUtil: No Webservice connection to migrate.
Mar 12, 2014 12:50:42 PM oracle.adfmf.framework.dt.migration.ApplicationMigrator _migrateApplication
INFO: Mobile application migration: granted all Device Access permissions in adfmf-application.xml. You may
revoke permissions not required by your application.
Mar 12, 2014 12:50:43 PM oracle.adfmf.framework.dt.migration.ApplicationMigrator _migrateFeature
INFO: Mobile application migration: migrated credentials for
C:\JDeveloper\mywork\mobileApp1\ViewController\src\META-INF\adfmf-feature.xml.
Mar 12, 2014 12:50:43 PM oracle.adfmf.framework.dt.migration.ApplicationMigrator
_addAuthenticationModeInConnections
INFO: Mobile application migration: migrated connections.xml to add authenticationMode where necessary.
Mar 12, 2014 12:50:43 PM oracle.adfmf.framework.dt.migration.ApplicationMigrator _migrateSyncXml
INFO: Mobile application migration: added sync-config.xml to the application.
Migration successfully completed for the following file(s):
C:\JDeveloper\mywork\mobileApp1\mobileApp1.jws
C:\JDeveloper\mywork\mobileApp1\ApplicationController\ApplicationController.jpr
C:\JDeveloper\mywork\mobileApp1\ViewController\ViewController.jpr
[12:50:44 PM] Migration finished.

3.4.1 What Happens When You Migrate an ADF Mobile Application

Table 3–1 describes how migration affects ADF Mobile artifacts.

File Name	Change		
adfmf-feature.xml	The migration makes the following changes:		
	 Renames the file as maf-feature.xml. 		
	 Replaces the credentials attribute with securityEnabled=true. 		
	 Transcribes the credentials attribute definition (defined as either local or remote) as a hybrid connection definition (<authenticationmode value="hybrid"></authenticationmode>) in the connections.xml file. 		
adfmf-application.xml	The migration renames the file as maf-application.xml.		

 Table 3–1
 Migration of ADF Mobile Artifacts and Configuration

File Name	Change			
connections.xml	The migration removes the secure SOAP web service connections defined by the <policy-references> element from the connections.xml file. These definitions are populated to the wsm-assembly.xml file. The migration creates stub connections.xml and wsm-assembly.xml files if the ADF Mobile application does not include a connections.xml file. If the ADF Mobile application includes a connections.xml that has no web services policy definitions, then the migration creates a stub wsm-assembly file.</policy-references>			
adfmf-config.xml	The migration renames the file as maf-config.xml. It also adds the default skin version for the skin family if the skin family is the default skin family and the skin version is not specified. For example, the maf-config.xml may be modified to include the following values:			
	<skin-family>mobileAlta</skin-family> <skin-version>v1.1</skin-version>			
adfmf-skins.xml	The migration renames the file as maf-skins.xml.			

Table 3–1 (Cont.) Migration of ADF Mobile Artifacts and Configuration

The application migrates from the ADF Mobile Framework technology to use the Mobile Application Framework technology as a project feature. Figure 3–4 shows the Features page for an application controller project that uses the Mobile Application Framework technology. Choose **Project Properties > Features** to view this dialog.

Figure 3–4 Mobile Application Framework Project Feature



MAF does not override the icon, splash screen, or navigation bar images created for the ADF Mobile application; the image files within the application controller's resources file are retained. Likewise, any images used for application features are also retained.

3.4.1.1 About Migrating Web Service Policy Definitions

MAF stores web service policy definitions in the wsm-assembly.xml file. ADF Mobile applications store this information in the connections.xml file. Example 3–1 illustrates oracle/wss_username_token_client_policy by the <policy-references> element in the connections.xml file.

Example 3–1 The connections.xml File

Example 3–2 illustrates the policy defined in the wsm-assembly.xml file.

Example 3–2 The wsm-assembly.xml File

```
<wsp:PolicyReference xmlns:wsp="http://www.w3.org/ns/ws-policy"
DigestAlgorithm="http://www.w3.org/ns/ws-policy/Sha1Exc"
URI="oracle/wss_username_token_client_policy"
orawsp:status="enabled"
orawsp:id="2"/>
```

3.4.2 What You May Need to Know About FARs in Migrated Applications

MAF does not migrate the adfmf-feature.xml file packaged within a Feature Archive (FAR) file. You replace the ADF Mobile FARs used by a migrated application to make sure that the credentials attribute has been replaced by securityEnabled=true in the FAR's maf-feature.xml file.

After you migrate the application:

- 1. Choose Application Properties > Libraries and Classpath.
- 2. Select the FAR and click **Remove**.
- **3.** Import the FAR containing the migrated view controller.
- **4.** Migrate the ADF Mobile application that contains the view controller project that was packaged as a FAR.

Note: A FAR cannot include both an adfmf-feature.xml file and a maf-feature.xml file.

- **1.** Deploy the view controller project as a FAR.
- 2. Import the FAR into the migrated application.

For more information about how to import a FAR into an application, see the "How to Use FAR Content in a MAF Application" section of *Developing Mobile Applications with Oracle Mobile Application Framework.*

3.5 Migrating to New cacerts File for SSL in MAF 2.1.0

MAF 2.1.0 delivers a new cacerts file for use in MAF applications. Make sure that the cacerts file packaged in the MAF application that you publish for your end users to

install contains the same CA root certificates as the HTTPS server that end users connect to when they use your MAF application.

You may need to import new certificates to your MAF application's cacerts file if the HTTPS server contains certificates not present in your MAF application's cacerts file. Similarly, system administrators for the HTTPS servers that your MAF application connects to may need to import new certificates if your MAF application uses a certificate not present on the HTTPS server.

Use JDK 8's keytool utility to view and manage the certificates in your MAF application's cacerts file. The following example demonstrates how you might use JDK 8's keytool utility to display the list of certificates in a cacerts file:

*JDK8install/*bin/keytool -list -v -keystore *dirPathToCacertsFile*/cacerts -storepass changeit | grep "Issuer:"

For more information about using the JDK 8's keytool utility to manage certificates, see http://docs.oracle.com/javase/8/docs/technotes/tools/#security. For example, to use the keytool utility on Windows, see

http://docs.oracle.com/javase/8/docs/technotes/tools/windows/keytool.html.
For UNIX-based operating systems, see

http://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html.

For more information about the cacerts file and using SSL to secure your MAF application, see the "Supporting SSL" section in *Developing Mobile Applications with Oracle Mobile Application Framework*.

Example 3–3 lists the issuers of CA root certificates included in MAF 2.1.0's cacerts file. Use JDK 8's keytool utility, as previously described, to manage the certificates in this file to meet the requirements of the environment where your MAF application will be used.

Example 3–3 CA Root Certificate Issuers in MAF 2.1.0 cacerts File

Issuer: CN=DigiCert Assured ID Root CA, OU=www.digicert.com, O=DigiCert Inc, C=US Issuer: CN=TC TrustCenter Class 2 CA II, OU=TC TrustCenter Class 2 CA, O=TC TrustCenter GmbH, C=DE Issuer: EMAILADDRESS=premium-server@thawte.com, CN=Thawte Premium Server CA, OU=Certification Services Division, O=Thawte Consulting cc, L=Cape Town, ST=Western Cape, C=ZA Issuer: CN=SwissSign Platinum CA - G2, O=SwissSign AG, C=CH Issuer: CN=SwissSign Silver CA - G2, O=SwissSign AG, C=CH Issuer: EMAILADDRESS=server-certs@thawte.com, CN=Thawte Server CA, OU=Certification Services Division, O=Thawte Consulting cc, L=Cape Town, ST=Western Cape, C=ZA Issuer: CN=Equifax Secure eBusiness CA-1, O=Equifax Secure Inc., C=US Issuer: CN=SecureTrust CA, O=SecureTrust Corporation, C=US Issuer: CN=UTN-USERFirst-Client Authentication and Email, OU=http://www.usertrust.com, O=The USERTRUST Network, L=Salt Lake City, ST=UT, C=US Issuer: EMAILADDRESS=personal-freemail@thawte.com, CN=Thawte Personal Freemail CA, OU=Certification Services Division, O=Thawte Consulting, L=Cape Town, ST=Western Cape, C=ZA Issuer: CN=AffirmTrust Networking, O=AffirmTrust, C=US Issuer: CN=Entrust Root Certification Authority, OU="(c) 2006 Entrust, Inc.", OU=www.entrust.net/CPS is incorporated by reference, O="Entrust, Inc.", C=US Issuer: CN=UTN-USERFirst-Hardware, OU=http://www.usertrust.com, O=The USERTRUST Network, L=Salt Lake City, ST=UT, C=US Issuer: CN=Certum CA, O=Unizeto Sp. z o.o., C=PL Issuer: CN=AddTrust Class 1 CA Root, OU=AddTrust TTP Network, O=AddTrust AB, C=SE Issuer: CN=Entrust Root Certification Authority - G2, OU="(c) 2009 Entrust, Inc. - for authorized use only", OU=See www.entrust.net/legal-terms, O="Entrust, Inc.", C=US Issuer: OU=Equifax Secure Certificate Authority, O=Equifax, C=US Issuer: CN=QuoVadis Root CA 3, O=QuoVadis Limited, C=BM Issuer: CN=QuoVadis Root CA 2, O=QuoVadis Limited, C=BM Issuer: CN=DigiCert High Assurance EV Root CA, OU=www.digicert.com, O=DigiCert Inc, C=US

Issuer: EMAILADDRESS=info@valicert.com, CN=http://www.valicert.com/, OU=ValiCert Class 1 Policy Validation Authority, O="ValiCert, Inc.", L=ValiCert Validation Network Issuer: CN=Equifax Secure Global eBusiness CA-1, O=Equifax Secure Inc., C=US Issuer: CN=GeoTrust Universal CA, O=GeoTrust Inc., C=US Issuer: OU=Class 3 Public Primary Certification Authority, O="VeriSign, Inc.", C=US Issuer: CN=thawte Primary Root CA - G3, OU="(c) 2008 thawte, Inc. - For authorized use only", OU=Certification Services Division, O="thawte, Inc.", C=US Issuer: CN=thawte Primary Root CA - G2, OU="(c) 2007 thawte, Inc. - For authorized use only", O="thawte, Inc.", C=US Issuer: CN=Deutsche Telekom Root CA 2, OU=T-TeleSec Trust Center, O=Deutsche Telekom AG, C=DE Issuer: CN=Buypass Class 3 Root CA, O=Buypass AS-983163327, C=NO Issuer: CN=UTN-USERFirst-Object, OU=http://www.usertrust.com, O=The USERTRUST Network, L=Salt Lake City, ST=UT, C=US Issuer: CN=GeoTrust Primary Certification Authority, O=GeoTrust Inc., C=US Issuer: CN=Buypass Class 2 Root CA, O=Buypass AS-983163327, C=NO Issuer: CN=Baltimore CyberTrust Code Signing Root, OU=CyberTrust, O=Baltimore, C=IE Issuer: OU=Class 1 Public Primary Certification Authority, O="VeriSign, Inc.", C=US Issuer: CN=Baltimore CyberTrust Root, OU=CyberTrust, O=Baltimore, C=IE Issuer: OU=Starfield Class 2 Certification Authority, O="Starfield Technologies, Inc.", C=US Issuer: CN=Chambers of Commerce Root, OU=http://www.chambersign.org, O=AC Camerfirma SA CIF A82743287, C=EU Issuer: CN=T-TeleSec GlobalRoot Class 3, OU=T-Systems Trust Center, O=T-Systems Enterprise Services GmbH. C=DE Issuer: CN=VeriSign Class 3 Public Primary Certification Authority - G5, OU="(c) 2006 VeriSign, Inc. - For authorized use only", OU=VeriSign Trust Network, O="VeriSign, Inc.", C=US Issuer: CN=T-TeleSec GlobalRoot Class 2, OU=T-Systems Trust Center, O=T-Systems Enterprise Services GmbH, C=DE Issuer: CN=TC TrustCenter Universal CA I, OU=TC TrustCenter Universal CA, O=TC TrustCenter GmbH, C=DE Issuer: CN=VeriSign Class 3 Public Primary Certification Authority - G4, OU="(c) 2007 VeriSign, Inc. - For authorized use only", OU=VeriSign Trust Network, O="VeriSign, Inc.", C=US Issuer: CN=VeriSign Class 3 Public Primary Certification Authority - G3, OU="(c) 1999 VeriSign, Inc. - For authorized use only", OU=VeriSign Trust Network, O="VeriSign, Inc.", C=US Issuer: CN=XRamp Global Certification Authority, O=XRamp Security Services Inc, OU=www.xrampsecurity.com, C=US Issuer: CN=Class 3P Primary CA, O=Certplus, C=FR Issuer: CN=Certum Trusted Network CA, OU=Certum Certification Authority, O=Unizeto Technologies S.A., C=PL Issuer: OU=VeriSign Trust Network, OU="(c) 1998 VeriSign, Inc. - For authorized use only", OU=Class 3 Public Primary Certification Authority - G2, O="VeriSign, Inc.", C=US Issuer: CN=GlobalSign, O=GlobalSign, OU=GlobalSign Root CA - R3 Issuer: CN=UTN - DATACorp SGC, OU=http://www.usertrust.com, O=The USERTRUST Network, L=Salt Lake City, ST=UT, C=US Issuer: OU=Security Communication RootCA2, O="SECOM Trust Systems CO.,LTD.", C=JP Issuer: CN=GTE CyberTrust Global Root, OU="GTE CyberTrust Solutions, Inc.", O=GTE Corporation, C=US Issuer: OU=Security Communication RootCA1, O=SECOM Trust.net, C=JP Issuer: CN=AffirmTrust Commercial, O=AffirmTrust, C=US Issuer: CN=TC TrustCenter Class 4 CA II, OU=TC TrustCenter Class 4 CA, O=TC TrustCenter GmbH, C=DE Issuer: CN=VeriSign Universal Root Certification Authority, OU="(c) 2008 VeriSign, Inc. - For authorized use only", OU=VeriSign Trust Network, O="VeriSign, Inc.", C=US Issuer: CN=GlobalSign, O=GlobalSign, OU=GlobalSign Root CA - R2 Issuer: CN=Class 2 Primary CA, O=Certplus, C=FR Issuer: CN=DigiCert Global Root CA, OU=www.digicert.com, O=DigiCert Inc, C=US Issuer: CN=GlobalSign Root CA, OU=Root CA, O=GlobalSign nv-sa, C=BE Issuer: CN=thawte Primary Root CA, OU="(c) 2006 thawte, Inc. - For authorized use only", OU=Certification Services Division, O="thawte, Inc.", C=US Issuer: CN=Starfield Root Certificate Authority - G2, O="Starfield Technologies, Inc.", L=Scottsdale, ST=Arizona, C=US Issuer: CN=GeoTrust Global CA, O=GeoTrust Inc., C=US Issuer: CN=Sonera Class2 CA, O=Sonera, C=FI

Issuer: CN=Thawte Timestamping CA, OU=Thawte Certification, O=Thawte, L=Durbanville, ST=Western Cape, C=ZA Issuer: CN=Sonera Class1 CA, O=Sonera, C=FI Issuer: CN=QuoVadis Root Certification Authority, OU=Root Certification Authority, O=QuoVadis Limited, C=BM Issuer: CN=AffirmTrust Premium ECC, O=AffirmTrust, C=US Issuer: CN=Starfield Services Root Certificate Authority - G2, O="Starfield Technologies, Inc.", L=Scottsdale, ST=Arizona, C=US Issuer: EMAILADDRESS=info@valicert.com, CN=http://www.valicert.com/, OU=ValiCert Class 2 Policy Validation Authority, O="ValiCert, Inc.", L=ValiCert Validation Network Issuer: CN=AAA Certificate Services, O=Comodo CA Limited, L=Salford, ST=Greater Manchester, C=GB Issuer: CN=America Online Root Certification Authority 2, O=America Online Inc., C=US Issuer: CN=AddTrust Qualified CA Root, OU=AddTrust TTP Network, O=AddTrust AB, C=SE Issuer: CN=KEYNECTIS ROOT CA, OU=ROOT, O=KEYNECTIS, C=FR Issuer: CN=America Online Root Certification Authority 1, O=America Online Inc., C=US Issuer: CN=VeriSign Class 2 Public Primary Certification Authority - G3, OU="(c) 1999 VeriSign, Inc. - For authorized use only", OU=VeriSign Trust Network, O="VeriSign, Inc.", C=US Issuer: CN=AddTrust External CA Root, OU=AddTrust External TTP Network, O=AddTrust AB, C=SE Issuer: OU=VeriSign Trust Network, OU="(c) 1998 VeriSign, Inc. - For authorized use only", OU=Class 2 Public Primary Certification Authority - G2, O="VeriSign, Inc.", C=US Issuer: CN=GeoTrust Primary Certification Authority - G3, OU=(c) 2008 GeoTrust Inc. - For authorized use only, O=GeoTrust Inc., C=US Issuer: CN=GeoTrust Primary Certification Authority - G2, OU=(c) 2007 GeoTrust Inc. - For authorized use only, O=GeoTrust Inc., C=US Issuer: CN=SwissSign Gold CA - G2, O=SwissSign AG, C=CH Issuer: CN=Entrust.net Certification Authority (2048), OU=(c) 1999 Entrust.net Limited, OU=www.entrust.net/CPS_2048 incorp. by ref. (limits liab.), O=Entrust.net Issuer: OU=ePKI Root Certification Authority, O="Chunghwa Telecom Co., Ltd.", C=TW Issuer: CN=Global Chambersign Root - 2008, O=AC Camerfirma S.A., SERIALNUMBER=A82743287, L=Madrid (see current address at www.camerfirma.com/address), C=EU Issuer: CN=Chambers of Commerce Root - 2008, O=AC Camerfirma S.A., SERIALNUMBER=A82743287, L=Madrid (see current address at www.camerfirma.com/address), C=EU Issuer: OU=Go Daddy Class 2 Certification Authority, O="The Go Daddy Group, Inc.", C=US Issuer: CN=AffirmTrust Premium, O=AffirmTrust, C=US Issuer: CN=VeriSign Class 1 Public Primary Certification Authority - G3, OU="(c) 1999 VeriSign, Inc. - For authorized use only", OU=VeriSign Trust Network, O="VeriSign, Inc.", C=US Issuer: OU=Security Communication EV RootCA1, O="SECOM Trust Systems CO.,LTD.", C=JP Issuer: OU=VeriSign Trust Network, OU="(c) 1998 VeriSign, Inc. - For authorized use only", OU=Class 1 Public Primary Certification Authority - G2, O="VeriSign, Inc.", C=US Issuer: CN=Go Daddy Root Certificate Authority - G2, O="GoDaddy.com, Inc.", L=Scottsdale, ST=Arizona, C=US