

Oracle® Database Mobile Server

Installation Guide

Release 12.1.0

E58647-01

January 2015

Copyright © 2014, 2015, Oracle and/or its affiliates. All rights reserved.

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

Contents

Preface	ix
Intended Audience.....	ix
Documentation Accessibility	ix
Related Documents	ix
Conventions	ix
1 Oracle Database Mobile Server Concepts	
1.1 Overview of Oracle Database Mobile Server.....	1-1
1.1.1 What is a Mobile Architecture?	1-1
1.1.2 What Are the Benefits of a Mobile Architecture?	1-1
1.1.3 Why Use Oracle Database Mobile Server?.....	1-2
1.2 Roadmap for Understanding and Using the Mobile Server.....	1-2
1.2.1 Designing and Developing your Mobile Applications	1-3
1.2.2 Synchronizing Data from a Mobile Client to the Oracle Database.....	1-3
1.2.2.1 Using SQLite Database	1-3
1.2.2.2 Using Berkeley DB Database	1-4
1.2.2.3 Using Java DB Database	1-4
1.2.3 Configuring and Managing the Mobile Server Environment.....	1-4
1.2.4 Performance Techniques	1-4
2 Third Party Licensing in Oracle Database Mobile Server	
2.1 SQLite Database	2-1
2.2 Oracle JDK 1.7, and JDK 1.8	2-1
2.3 Third Party Licensing for ZLib and JZLib.....	2-1
3 Requirements Before Installation or Development	
3.1 Include the Latest Patches	3-1
3.2 Release Notes.....	3-1
3.3 Supported Components and Technologies.....	3-1
3.3.1 Certified Oracle RDBMS to Use With Oracle Database Mobile Server	3-2
3.3.2 JDK Platform Support	3-2
3.3.3 Certified Application Server Configurations.....	3-3
3.3.4 Certified Platforms and Technologies for the Mobile Server.....	3-3
3.3.5 Certified Platforms for the Mobile Development Kit.....	3-3
3.3.6 Certified Versions for OID Support.....	3-4

3.3.7	Certified Browsers	3-4
3.4	Installation Requirements for the Oracle Database and Application Server in Working With Oracle Database Mobile Server	3-4
3.4.1	Installation Requirements for the Oracle Database for Working With a Mobile Client	3-4
3.4.2	Installation Requirements for Using an Application Server With Oracle Database Mobile Server	3-4
3.4.2.1	Installing Oracle WebLogic Server	3-5
3.4.2.2	Installing Oracle GlassFish Server Version 3.1.....	3-5
3.4.2.3	Installing GlassFish Server Open Source Edition	3-5
3.4.2.4	Installing Apache TomEE 1.6 Web Profile.....	3-5
3.5	Hardware Requirements for Mobile Server on Windows	3-5
3.6	System Requirements For UNIX Systems	3-6
3.7	Recommended System Configuration For UNIX.....	3-6
3.7.1	Setting UNIX Environment Variables	3-6
3.7.1.1	ORACLE_HOME and MOBILE_HOME.....	3-7
3.7.1.2	JDK_HOME and PATH.....	3-7
3.7.1.3	DISPLAY	3-8
3.7.1.4	TMP and TMPDIR.....	3-8
3.7.2	Pre-Install Requirements Before Installing the MDK on LINUX.....	3-9
3.7.3	Creating UNIX Accounts and Groups.....	3-9
3.7.3.1	UNIX Group Name For the Oracle Universal Installer Inventory	3-9
3.7.3.2	UNIX Account to Own Oracle Software	3-9
3.7.4	Configuring Kernel Parameters and Shell Limits for UNIX	3-10
3.7.4.1	Configuring Shell Limits and System Configuration Parameters on AIX	3-10
3.7.4.2	Configuring Kernel Parameters on HP-UX	3-11
3.7.4.3	Configuring the Kernel Parameters on Linux	3-11
3.7.4.4	Set Shell Limits for the User oracle	3-12
3.8	System Requirements for Mobile Clients	3-13
3.9	Setting Up Location of the Datafile on the Server.....	3-13
3.10	Defining Synchronization Tablespace	3-13

4 Installation of Oracle Database Mobile Server

4.1	Oracle Database Mobile Server Installation Considerations	4-1
4.1.1	Installing Multiple Languages on a Single Solaris Machine	4-1
4.1.2	National Language Support for Chinese, Japanese, and Korean	4-1
4.1.3	Mobile Server on a DHCP Server is Not Supported.....	4-1
4.1.4	Installing Mobile Server Using Multiple Languages.....	4-2
4.2	Starting Oracle Universal Installer	4-2
4.2.1	Starting the Oracle Universal Installer on Windows.....	4-2
4.2.2	Starting Oracle Universal Installer on UNIX	4-3
4.2.2.1	Using the oraInventory Directory	4-3
4.2.2.2	Location of Files on UNIX	4-4
4.3	Installing Oracle Database Mobile Server	4-4
4.3.1	GUI Install of the Mobile Server and MDK Components.....	4-4
4.3.1.1	Installation of the Mobile Server	4-5
4.3.1.2	Installation of Mobile Development Kit.....	4-21

4.3.2	Silent Install of the Mobile Server and MDK Components	4-22
4.3.2.1	Silent Install for Mobile Server or Mobile Development Kit on Windows	4-22
4.3.2.2	Silent Install Mobile Server or the Mobile Development Kit on UNIX	4-24
4.3.2.3	Configuration File for Deploying Mobile Serve.....	4-26
4.3.3	Providing High Availability with a Farm of Multiple Mobile Servers	4-28
4.3.4	Install Oracle Database Mobile Server on Linux	4-29
4.3.4.1	Patch Required if Using Linux Redhat 3.0.....	4-29
4.3.4.2	Providing Enough Swap Space on the Linux Platform	4-29
4.3.5	Mobile Client Install	4-29
4.4	Starting and Stopping Mobile Server.....	4-29
4.4.1	Starting Mobile Server	4-30
4.4.2	Stopping Mobile Server	4-30
4.5	Testing Your Mobile Server Installation.....	4-31
4.6	Removing Demo Applications.....	4-32
4.7	How to Uninstall Oracle Database Mobile Server	4-32

5 Upgrade Oracle Database Mobile Server

5.1	Upgrade Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11g.....	5-1
5.1.1	Pre-Upgrade Instructions	5-2
5.1.2	Upgrade Mobile Server	5-2
5.1.2.1	Upgrade in the same ORACLE_HOME.....	5-4
5.1.2.2	Upgrade in a separate ORACLE_HOME.....	5-4
5.1.2.3	Upgrade Your Mobile Server Repository	5-5
5.1.2.4	Upgrade Your Existing Applications	5-5
5.1.2.5	Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory (OID)	5-6
5.1.3	Mobile Server Farm Upgrade	5-6
5.1.4	Upgrade Mobile Client	5-6
5.1.4.1	Upgrade Oracle Mobile Clients from De-Supported Mobile Client Platforms... ..	5-8
5.2	Upgrade Oracle Database Mobile Server 11.1.0 or 11.2.0 to 11.3.0.....	5-8
5.2.1	Upgrade Mobile Server	5-9
5.2.1.1	Upgrading in the Same ORACLE_HOME.....	5-9
5.2.1.2	Upgrading in a Separate ORACLE_HOME.....	5-9
5.2.1.3	Upgrade Your Mobile Server Repository	5-10
5.2.1.4	Upgrade Your Existing Applications	5-10
5.2.1.5	Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory (OID)	5-10
5.2.2	Upgrade Mobile Client	5-10
5.2.2.1	Upgrade Berkeley DB Mobile Client to 11.3.0.....	5-11
5.2.3	Mobile Server Farm Upgrade	5-11
5.3	Upgrade Oracle Database Mobile Server 11.1.0 to 11.2.0.....	5-12
5.3.1	Mobile Server Farm Upgrade	5-12
5.3.2	Upgrade Oracle Database Mobile Server 11.1.0.0.0 to 11.2.0.0.0.....	5-12
5.3.2.1	Upgrade in the Same or a Separate ORACLE_HOME	5-13
5.3.2.2	Upgrade Your Mobile Server Repository	5-14
5.3.3	Upgrade Your Existing Applications.....	5-14
5.3.4	Upgrade Mobile Clients.....	5-14

5.4	Upgrade Oracle Database Lite from 10g Release 1 or 2 to 10g Release 3	5-15
5.4.1	Supported Versions for Upgrading Oracle Database Lite	5-15
5.4.1.1	Upgrading in the Same ORACLE_HOME.....	5-15
5.4.1.2	Upgrading in a Separate ORACLE_HOME.....	5-15
5.4.1.3	Supported Oracle Database Lite Versions for Upgrade	5-15
5.4.2	Pre-Upgrade Instructions	5-18
5.4.3	Upgrade Your Mobile Server Repository	5-18
5.4.4	Upgrade Your Existing Applications	5-18
5.4.5	Installing Sample Applications.....	5-19
5.4.6	Migrate Your Users From the Mobile Server Repository to the Oracle Internet Directory (OID)	5-20
5.4.6.1	Mobile Server Installed On Oracle Application Server 10.1.2.0.0.....	5-20
5.4.6.2	Mobile Server Installed on Oracle Application Server 10.1.3.1.0 or Higher	5-21
5.4.7	Remove the 10g Release 1 or 2 Installation	5-22
5.4.8	Remove Duplicate Mobile Server IDs from the Repository	5-22
5.5	Upgrade Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 12c	5-23
5.5.1	Pre-Upgrade Instructions	5-23
5.5.2	Upgrade Mobile Server.....	5-24
5.5.2.1	Upgrade in the same ORACLE_HOME.....	5-25
5.5.2.2	Upgrade in a separate ORACLE_HOME.....	5-25
5.5.2.3	Upgrade Your Mobile Server Repository	5-26
5.5.2.4	Upgrade Your Existing Applications	5-26
5.5.2.5	Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory (OID)	5-27
5.5.3	Mobile Server Farm Upgrade	5-27
5.5.4	Upgrade Mobile Client	5-27
5.5.4.1	Upgrade Oracle Mobile Clients from De-Supported Mobile Client Platforms	5-29
5.6	Upgrade Oracle Database Mobile Server 11g to 12.1.0.....	5-29
5.6.1	Upgrade Mobile Server.....	5-29
5.6.1.1	Upgrading in the Same ORACLE_HOME.....	5-30
5.6.1.2	Upgrading in a Separate ORACLE_HOME.....	5-30
5.6.1.3	Upgrade Your Mobile Server Repository	5-30
5.6.1.4	Upgrade Your Existing Applications	5-30
5.6.1.5	Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory (OID)	5-31
5.6.2	Mobile Server Farm Upgrade	5-31
5.6.3	Upgrade Mobile Client	5-31
5.6.3.1	Upgrade Berkeley DB Mobile Client	5-32

6 Migrating from Oracle Lite Client Database to Berkeley DB

6.1	Migration Guideline	6-1
6.1.1	Backup OLite Database.....	6-2
6.1.2	Upgrade Oracle Lite Mobile Client	6-2
6.1.3	Export Oracle Lite Database.....	6-2
6.1.4	Migrate OLite Publication	6-3
6.1.5	Upgrade Oracle Database Lite Mobile Server 10.3.0.3.0 to Oracle Database Mobile Server 12c	6-3
6.1.6	Create Berkeley DB-based Publication and Application	6-4

6.1.7	Install Berkeley DB Mobile Client	6-4
6.1.8	Import Data to Berkeley DB	6-4

Index

Preface

This preface introduces you to the *Oracle Database Mobile Server Installation Guide*, discussing the intended audience, documentation accessibility, related documents, and conventions of this document.

Intended Audience

This manual is intended for users, who are new to the product, and administrators who are installing or upgrading Oracle Database Mobile Server.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

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

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

Use the following manuals as reference when installing and configuring OracleAS, WebLogic or Glassfish server:

- *Oracle® Fusion Middleware Administrator's Guide*
- *Oracle® Fusion Middleware Installation Guide for Oracle WebLogic Server*
- *Oracle® GlassFish Server 3.1 Installation Guide*

Conventions

The following conventions are also used in this manual:

Convention	Meaning
.	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
.	
.	

Convention	Meaning
...	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted
boldface text	Boldface type in text indicates a term defined in the text, the glossary, or in both locations.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.
<i>italic monospace</i>	Italic monospace type indicates a variable in a code example that you must replace. For example: <pre>Driver=<i>install_dir</i>/lib/libtten.sl</pre> Replace <i>install_dir</i> with the path of your TimesTen installation directory.
<>	Angle brackets enclose user-supplied names.
[]	Brackets enclose optional clauses from which you can choose one or none.

Oracle Database Mobile Server Concepts

The following sections provide an introduction to Oracle Database Mobile Server and its components:

- [Section 1.1, "Overview of Oracle Database Mobile Server"](#)
- [Section 1.2, "Roadmap for Understanding and Using the Mobile Server"](#)

1.1 Overview of Oracle Database Mobile Server

Oracle Database Mobile Server provides a complete mobile infrastructure designed to run enterprise database applications for clients using either Berkeley DB, SQLite, or Java DB. Oracle Database Mobile Server provides the infrastructure that makes the enterprise application and data store available even when communications to the enterprise itself are not available or reliable. Oracle Database Mobile Server brings the applications that were once limited to the base office and deploys these applications out into the world where they are most needed.

The following sections describe how you can use Oracle Database Mobile Server to streamline your operations in the field:

- [Section 1.1.1, "What is a Mobile Architecture?"](#)
- [Section 1.1.2, "What Are the Benefits of a Mobile Architecture?"](#)
- [Section 1.1.3, "Why Use Oracle Database Mobile Server?"](#)

1.1.1 What is a Mobile Architecture?

The mobile architecture completes the enterprise system by merging the enterprise infrastructure with every remote aspect of the organization. A mobile architecture contains the remote application, the remote database, and the remote rules of the business. The Oracle Database Mobile Server mobile infrastructure is responsible for connecting and synchronizing applications, associated data, and business rules with the applications, database, and business rules of the enterprise.

1.1.2 What Are the Benefits of a Mobile Architecture?

A mobile architecture with the proper design, security components, and implementation saves money. Oracle Database Mobile Server can remove some of the manual processes performed out in the field. In the past, you may have manually written down the information in the field and then manually entered the data in the enterprise database once you returned to the corporate environment. With Oracle Database Mobile Server, you can capture the data once in the field by entering the data into a client device. Then, this data is synchronized up to the enterprise without

returning to the office to manually enter data. This removes a loss of productivity due to manual processes and sends the data immediately to the enterprise where it belongs. In addition, data can flow bi-directionally. If you need information at the remote site that has been updated at the office, this data is brought down to the client device during synchronization.

An application is created, where the user enters data on a client device, known as the mobile client. The mobile client can use either Berkeley DB, SQLite or Java DB database as the client database. The data on the mobile client is synchronized with a back-end Oracle database. For example, if you have a sales force, each sales person retrieves only his/her data on the client device. Any modifications made on the mobile client by the sales person in regards to his/her accounts or modified on the server by the office can be synchronized.

1.1.3 Why Use Oracle Database Mobile Server?

Oracle Database Mobile Server provides a complete mobile infrastructure suitable for almost any enterprise demands using the following:

- The mobile server repository resides in the back-end enterprise database, which links the enterprise data with the mobile data.
- The mobile server is a Web-based tier that integrates with Oracle WebLogic Server, Oracle GlassFish, or Apache TomEE.

This accesses remote locations through different types of wireless or wired connectivity. It facilitates the major functions for the mobile option, such as synchronization, application management, device management, and so on.

- The mobile client uses a client database, which can be Berkeley DB, SQLite, or Java DB database, and the means for deploying applications developed using the most popular languages. The mobile client database stores the relational data in one or more data files on the file system on the client. While the SQLite database is already installed on many client devices, you can install the Berkeley DB database on most any device from a cell phone, to a personal digital assistant (PDA), Tablet PC, Laptop, and so on from the Mobile Manager.

1.2 Roadmap for Understanding and Using the Mobile Server

The Database Mobile Server is a compilation of two products: the mobile server and the mobile client.

In an enterprise system, the mobile server facilitates the synchronization of data between multiple mobile clients and Oracle databases. The mobile client facilitates the synchronization of data from an independently installed client database to the mobile server. The supported client databases are listed below:

- *SQLite database*: If you are using the SQLite database as the client database, you do not need to install the SQLite database independently when using a native mobile client, but you must install it independently when using pure Java mobile client. If it is not already installed, refer to <http://www.sqlite.org/> for more information on installation, configuration and usage.
- *Berkeley DB*: If you are using the Berkeley DB as the client database, you do not need to install the Berkeley DB database independently when using a native mobile client, but you must install it independently when using pure Java mobile client. If it is not already installed, refer to <http://www.oracle.com/technology/products/berkeley-db/index.html> for more information on installation, configuration, and usage.

- *Java DB database*: Java DB is bundled with Oracle JDKs. If you are using Java DB as client database, you can only use pure Java Sync Engine (pure Java SE). Refer to <http://db.apache.org/derby> for more information on configuring, managing and using Java DB.

In a mobile environment, install the mobile client to enable synchronization. To properly install, develop and administer all components in the mobile server, first follow instructions in the mobile server books.

The installation of a mobile client includes the Sync Engine for managing synchronization between the client database and the back-end Oracle database. For details, see Chapter 2, "Installing the Mobile Client" in the *Oracle Database Mobile Server Mobile Client Guide*.

The following sections provide a roadmap for where to find the information to design, develop and manage the mobile server:

- [Section 1.2.1, "Designing and Developing your Mobile Applications"](#)
- [Section 1.2.2, "Synchronizing Data from a Mobile Client to the Oracle Database"](#)
- [Section 1.2.3, "Configuring and Managing the Mobile Server Environment"](#)
- [Section 1.2.4, "Performance Techniques"](#)

1.2.1 Designing and Developing your Mobile Applications

When you design a mobile application that provides synchronization of data from several users on multiple devices to one or more Oracle databases, you must carefully consider a balance between application functionality and minimizing the data being synchronized before you start to develop your application. Synchronization from multiple users and devices can cause performance issues. Thus, the design has an impact on how performant is the data synchronization.

To streamline your development, we have provided several sections guiding you through design decisions before developing a mobile application. For more information, see Chapter 1, "Overview for Designing Mobile Applications" in the *Oracle Database Mobile Server Developer's Guide*.

1.2.2 Synchronizing Data from a Mobile Client to the Oracle Database

The following sections describe each mobile client database type which you can use to synchronize data to the back-end Oracle database:

- [Section 1.2.2.1, "Using SQLite Database"](#)
- [Section 1.2.2.2, "Using Berkeley DB Database"](#)
- [Section 1.2.2.3, "Using Java DB Database"](#)

1.2.2.1 Using SQLite Database

The SQLite database is a small, compact, and self-contained database available on multiple platforms and available to the public. It has a small footprint and is easy to install and administer. In addition, many devices already have the SQLite database installed, including Android and Blackberry devices.

The SQLite database can be used with the mobile server for synchronizing data to a back-end Oracle Database or an Oracle RAC database. For Win32, WinCE, Linux, Android, and iOS platforms, both automatic and manual synchronization are supported. On Blackberry devices, only manual synchronization is supported. For

more information, see Chapter 2, "Installing the Mobile Client" in the *Oracle Database Mobile Server Mobile Client Guide*.

1.2.2.2 Using Berkeley DB Database

Berkeley DB is a small database that has a small footprint and is easy to administer. The mobile server gathers and synchronizes data from multiple mobile devices from several users.

Berkeley DB is its own product and is described completely in the *Berkeley DB documentation*. It can be used with the mobile server for synchronizing data to a back-end Oracle database or an Oracle RAC database. For Win32, WinCE, Linux, Android and iOS platforms, both automatic and manual synchronization are supported.

1.2.2.3 Using Java DB Database

Java DB is a full - featured, small footprint SQL database, which comes bundled with Oracle JDKs. Java DB is pure Java and it runs on Java SE platforms, that is platforms where Java SE runs.

Using the mobile client, which is installed independently, you can synchronize multiple Java DB databases with a back-end Oracle database.

If you want to use a Java DB database on Java SE platforms and synchronize the data entered in the Java DB database to a back-end Oracle database or an Oracle RAC database, download the pure Java Sync Engine archive file for Java DB, which supports both automatic and manual synchronization.

1.2.3 Configuring and Managing the Mobile Server Environment

The Mobile Manager is a GUI tool that enables you to configure and manage the mobile server. Some of the functions you can perform through the Mobile Manager are as follows:

- Create users.
- Manage permissions.
- Execute statistics.
- Determine performance of your SQL queries.
- Resolve errors, including conflict errors.

Configuration and management of the mobile server are described in the *Oracle Database Mobile Server Administration and Deployment Guide*.

1.2.4 Performance Techniques

Mobile devices do not have the processing power and memory that standard enterprise systems maintain. If the mobile applications and infrastructure are not tuned appropriately they are of little benefit to the organization.

The most important performance concepts for a mobile infrastructure are as follows:

- The time it takes to enter and retrieve data.
- The time it takes to synchronize data with the enterprise data store.

See *Oracle Database Mobile Server Troubleshooting and Tuning Guide* for techniques to enhance your performance for Oracle Database Mobile Server.

Third Party Licensing in Oracle Database Mobile Server

This chapter includes third-party license information for all third-party products included with Oracle Database Mobile Server. Oracle acknowledges that following Third Party and Open Source software are used in the provided programs covered by this documentation.

- [Section 2.1, "SQLite Database"](#)
- [Section 2.2, "Oracle JDK 1.7, and JDK 1.8"](#)
- [Section 2.3, "Third Party Licensing for ZLib and JZLib"](#)

2.1 SQLite Database

Any SQLite database services that are provided with Oracle Database Mobile Server are provided as a convenience to you and are provided "AS IS" with no express or implied conditions, endorsements, guarantees, representations, or warranties of any kind by Oracle and Oracle assumes no liability whatsoever, in relation thereto.

2.2 Oracle JDK 1.7, and JDK 1.8

This product includes code provided by Oracle.



2.3 Third Party Licensing for ZLib and JZLib

Oracle gratefully acknowledges the contributions of Jean-loup Gailly and Mark Adler, the authors of zlib.

This product includes JZlib.

Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Requirements Before Installation or Development

Before you install, you must check to see that you have the correct hardware and software necessary for using Oracle Database Mobile Server on your operating system. The requirements for each type of operating system are detailed in the following sections:

- [Section 3.1, "Include the Latest Patches"](#)
- [Section 3.2, "Release Notes"](#)
- [Section 3.3, "Supported Components and Technologies"](#)
- [Section 3.4, "Installation Requirements for the Oracle Database and Application Server in Working With Oracle Database Mobile Server"](#)
- [Section 3.5, "Hardware Requirements for Mobile Server on Windows"](#)
- [Section 3.6, "System Requirements For UNIX Systems"](#)
- [Section 3.7, "Recommended System Configuration For UNIX"](#)
- [Section 3.8, "System Requirements for Mobile Clients"](#)
- [Section 3.9, "Setting Up Location of the Datafile on the Server"](#)
- [Section 3.10, "Defining Synchronization Tablespace"](#)

3.1 Include the Latest Patches

For the latest information and patches, refer to Oracle *MetaLink* at the following website:

<http://metalink.oracle.com>

3.2 Release Notes

Read the *Oracle Database Mobile Server Release Notes* before installing Oracle Database Mobile Server, which are available as part of the documentation shipped with Oracle Database Mobile Server. The most up-to-date version is available at OTN at the following website:

<http://www.oracle.com/technetwork/documentation/index.html>

3.3 Supported Components and Technologies

The following sections detail the supported components and technologies:

- [Section 3.3.1, "Certified Oracle RDBMS to Use With Oracle Database Mobile Server"](#)
- [Section 3.3.2, "JDK Platform Support"](#)
- [Section 3.3.3, "Certified Application Server Configurations"](#)
- [Section 3.3.4, "Certified Platforms and Technologies for the Mobile Server"](#)
- [Section 3.3.5, "Certified Platforms for the Mobile Development Kit"](#)
- [Section 3.3.6, "Certified Versions for OID Support"](#)
- [Section 3.3.7, "Certified Browsers"](#)

3.3.1 Certified Oracle RDBMS to Use With Oracle Database Mobile Server

Use one of the following database versions with Oracle Database Mobile Server: Oracle Database 10g, Oracle Database 11g, or Oracle 12c Pluggable Database.

3.3.2 JDK Platform Support

Different JDKs are supported for mobile client, Mobile Server and Mobile Development Kit. For all non Java mobile clients, use 32-bit Oracle JDK 1.7 or 1.8. For pure Java mobile client and Mobile Development Kit, use 32-bit Oracle JDK 1.7 or 1.8 on 32-bit platform, and 64-bit Oracle JDK 1.7 or 1.8 on 64-bit platform. For the mobile server, the JDK version required depends on what target platform you are going to run the mobile server and what version of the application server you are using. Refer to [Table 3–1](#) for detailed information.

Table 3–1 *JDK Version Supported*

Application Server Version Used	JDK Version Supported
WebLogic Server 11g Release 1	JRockit JDK 1.6 or Oracle JDK 1.6 IBM J9 VM 1.7 for AIX HP-UX 11i JDK for HP-UX
WebLogic Server 12c Release 1 (12.1.1, 12.1.2 or 12.1.3)	Oracle JDK 1.7 or 1.8 IBM J9 VM 1.7 for AIX HP-UX 11i JDK for HP-UX
Oracle GlassFish 3.1	Oracle JDK 1.7 IBM J9 VM 1.7 for AIX HP-UX 11i JDK for HP-UX
GlassFish Server 4 Open Source Edition	Oracle JDK 1.7 or Oracle JDK 1.8 IBM J9 VM 1.7 for AIX HP-UX 11i JDK for HP-UX
Apache TomEE 1.7 Web Profile	Oracle JDK 1.7 or Oracle JDK 1.8 HP-UX 11i JDK for HP-UX

Install JDK before installing the mobile server or Mobile Development Kit.

You should install all of the patches required for the JDK version you are using on the Windows operating system. This is constantly under review and published on the JDK download page on the Oracle website.

The `JDK_HOME` environment variable must be set to the directory where the Java Development Kit has been installed. For more information on `JDK_HOME`, see [Section 3.7.1.2, "JDK_HOME and PATH"](#) for setting the `JAVA` environment variables on Linux.

3.3.3 Certified Application Server Configurations

The following lists accepted configuration options for a middle-tier application server:

- Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c Release 1 (12.1.1, 12.1.2 and 12.1.3)
- Oracle GlassFish Server 3.1. For more information, see the following site:
<http://www.oracle.com/technetwork/middleware/glassfish/documentation/index.html>
- GlassFish Server Open Source Edition 4.0. For more information, see the following site:
<https://glassfish.java.net/>
- Apache TomEE 1.7.0. For more information, see the following site:
<http://openejb.apache.org/apache-tomee.html>

3.3.4 Certified Platforms and Technologies for the Mobile Server

You can install the mobile server on the following platforms:

- Microsoft Windows 7 (32-bit and 64-bit)
- Microsoft Windows 2003 (64-bit)
- Microsoft Windows 2008 R2 (64-bit)
- Oracle Enterprise Linux 5.0, or 6.0 with or without Unbreakable Enterprise Kernel (32-bit and 64-bit)
- OpenSUSE 13.1 (64-bit)
- Ubuntu 14.04 (64-bit)
- Fedora 20 (64-bit)
- Solaris 10 (64-bit, SPARC) and Solaris 11 (64-bit, X86)
- AIX 6.1 (64-bit)
- Microsoft Windows 8 (64-bit)
- HP-UX Itanium 11i v3 (64-bit)

3.3.5 Certified Platforms for the Mobile Development Kit

The Mobile Development Kit (MDK) is certified for Oracle JDK 1.7 and 1.8 and can be installed on the following platforms:

- Microsoft Windows 2003 (64-bit)
- Microsoft Windows 2008 R2 (64-bit)
- Microsoft Windows 7 (32-bit and 64-bit)
- Microsoft Windows 8 (64-bit)

- Oracle Enterprise Linux 5.0, or 6.0 containing Unbreakable Enterprise Kernel (32-bit and 64-bit)
- Ubuntu 14.04 (64-bit)

3.3.6 Certified Versions for OID Support

The following versions of OID are supported in conjunction with the mobile server:

- OID 10.1.4.
- OID 11.0

3.3.7 Certified Browsers

You can use any of the latest modern browsers.

3.4 Installation Requirements for the Oracle Database and Application Server in Working With Oracle Database Mobile Server

The following sections describe installation requirements for the Oracle Database and the application server you choose to use:

- [Section 3.4.1, "Installation Requirements for the Oracle Database for Working With a Mobile Client"](#)
- [Section 3.4.2, "Installation Requirements for Using an Application Server With Oracle Database Mobile Server"](#)

3.4.1 Installation Requirements for the Oracle Database for Working With a Mobile Client

When you synchronize the mobile client, your changes are updated in an Oracle back-end database. Thus, you must have either the Express or Standard or Enterprise Edition Oracle database to use the synchronization ability of Oracle Database Mobile Server.

3.4.2 Installation Requirements for Using an Application Server With Oracle Database Mobile Server

Oracle Database Mobile Server uses a middle-tier application server to communicate between the mobile clients and the back-end Oracle database.

Install the appropriate application server before installing Oracle Database Mobile Server, which can be Oracle WebLogic Server 11g, Oracle WebLogic Server 12c, Oracle GlassFish 3.1, GlassFish Server Open Source Edition 4.0, or Apache TomEE 1.7.

Note: Oracle Database Mobile Server uses ISO-8859-1 for URL decoding. Ensure that the application server uses the same configuration. For example, you can start Weblogic with extra configuration options as below:

```
export JAVA_OPTIONS="{JAVA_OPTIONS}"
-Dweblogic.http.URIDecodeEncoding=ISO-8859-1"
${WLS_DOMAIN_DIR}/bin/startWebLogic.sh
```

Note: For more information about how Oracle Database Mobile Server works with the middle-tier and the back-end database, see Chapter 1, "Overview for Designing Mobile Applications" in the *Oracle Database Mobile Server Developer's Guide*.

The following sections provide additional information when installing certain application servers:

- [Installing Oracle WebLogic Server](#)
- [Installing Oracle GlassFish Server Version 3.1](#)
- [Installing GlassFish Server Open Source Edition](#)
- [Installing Apache TomEE 1.6 Web Profile](#)

3.4.2.1 Installing Oracle WebLogic Server

Make sure that you install Oracle WebLogic before installing Oracle Database Mobile Server.

Oracle WebLogic Server 12c Release 1 uses some updated Java APIs which are included in its own distribution. You must prepare the related jar files after installation of JRockit 1.6 and Oracle WebLogic Server 12c.

Follow the steps below:

1. Create directory structure of `$JAVA_HOME/jre/lib/endorsed`, where `JAVA_HOME` is the directory in which JRockit was installed.
2. Copy `$WL_HOME/endorsed/*.jar` to `$JAVA_HOME/jre/lib/endorsed`.

Note: If you fail to do the above preparation, the deployment of Mobile Server will run into an "Exception thrown by startServer: java.lang.reflect.InvocationTargetException" error. These steps are only required by WebLogic Server 12c.

3.4.2.2 Installing Oracle GlassFish Server Version 3.1

Ensure that you install Oracle GlassFish server before installing Oracle Database Mobile Server.

3.4.2.3 Installing GlassFish Server Open Source Edition

Ensure that you install GlassFish Server Open Source Edition 4.0 before installing Oracle Database Mobile Server.

3.4.2.4 Installing Apache TomEE 1.6 Web Profile

Ensure that you install Apache TomEE 1.6 web profile before installing Oracle Database Mobile Server.

3.5 Hardware Requirements for Mobile Server on Windows

Before you install the mobile server, you must check to see that you have the correct hardware necessary for your Windows machines that use Oracle Database Mobile Server.

The hardware requirements for each component of Oracle Database Mobile Server for Windows are described in the following table:

Table 3–2 Hardware Requirements for Windows

Component	Hardware Requirements for this Component
Oracle Database Mobile Server	CPU: Pentium 4, 3 GHz Disk Space: 1 GB RAM: 1 GB Swap Space: 1535 MB
Oracle Database Mobile Server using Oracle WebLogic Server	See the "Oracle Fusion Middleware System Requirements and Specifications" document for the specific release version in the Oracle Fusion Middleware documentation.
Oracle Database Mobile Server using Oracle GlassFish Server	See the "Hardware and Software Requirements in Oracle GlassFish Server 3.1 Release Notes" document in the Oracle Fusion Middleware documentation.
GlassFish Server Open Source Edition 4.0	See "Hardware and Software Requirements" in GlassFish Server Open Source Edition Release Notes.
Mobile Development Kit	CPU: Pentium 4, 3 GHz Disk Space: 512 MB RAM: 512 MB Swap Space: 1535 MB

Note: The requirements for mobile server is related to the number of users synchronizing and the amount of data transferred.

3.6 System Requirements For UNIX Systems

Before you install, you must check to see that you have the correct hardware and software that satisfy the minimum and general system requirements for the Web server on the UNIX machines that use Oracle Database Mobile Server. See the Web server documentation for these requirements.

3.7 Recommended System Configuration For UNIX

This section describes the following system configurations for UNIX-based systems:

- [Section 3.7.1, "Setting UNIX Environment Variables"](#)
- [Section 3.7.2, "Pre-Install Requirements Before Installing the MDK on LINUX"](#)
- [Section 3.7.3, "Creating UNIX Accounts and Groups"](#)
- [Section 3.7.4, "Configuring Kernel Parameters and Shell Limits for UNIX"](#)

3.7.1 Setting UNIX Environment Variables

The following sections describe the environment variables that must be set before starting Oracle Universal Installer:

- [Section 3.7.1.1, "ORACLE_HOME and MOBILE_HOME"](#)
- [Section 3.7.1.2, "JDK_HOME and PATH"](#)

- [Section 3.7.1.3, "DISPLAY"](#)
- [Section 3.7.1.4, "TMP and TMPDIR"](#)

Note: Ensure your `PATH`, `CLASSPATH` and library path environment variables do not exceed 1024 characters. Longer values might generate errors such as "Word too long" during installation.

Refer to [Table 3–3](#) for the name of the library path environment variable for your platform.

[Table 3–3](#) lists the names of the library path environment variables for each platform.

Table 3–3 Library Path Environment Variable

Platform	Library Path Environment Variable
Linux	LD_LIBRARY_PATH
Solaris	LD_LIBRARY_PATH
AIX	LIBPATH
HP-UX	SHLIB_PATH and LD_LIBRARY_PATH

3.7.1.1 ORACLE_HOME and MOBILE_HOME

The Oracle home directory is the `root` directory in which Oracle software is installed. For Oracle Database Mobile Server, you can also use `MOBILE_HOME` environment variable to specify the root directory in which you want to install Mobile Server or Mobile Development Kit. The `CLASSPATH` contains the entire path you enter for `ORACLE_HOME`; thus, the length of `ORACLE_HOME` effects the length of the `CLASSPATH`. There are limits to the length of the values of the `CLASSPATH` values with JDK. If the `ORACLE_HOME` path is long, this will result in a long `CLASSPATH` and might cause problems running Oracle Database Mobile Server. The workaround is to shorten `ORACLE_HOME` path.

The `ORACLE_HOME` environment variable must be set before starting the installer, which must be set to the directory where you want to install.

3.7.1.2 JDK_HOME and PATH

The Linux, Solaris, AIX and HP-UX platforms require the `JDK_HOME` environment variable be set to the directory where the Java Development Kit has been installed. If the JDK has not been installed, install it before proceeding with the installation of the Mobile Server or the Mobile Development Kit. For the appropriate JDK version, see [Section 3.3.2, "JDK Platform Support"](#).

Before installation of a Linux/Solaris/AIX/HP-UX mobile server and the Linux MDK, set `JDK_HOME` to the JDK home directory.

Initialize the `JDK_HOME` and `PATH` environment variables, as follows:

```
export JDK_HOME=/path/to/jdk
export PATH=$JDK_HOME/bin:$ORACLE_HOME/bin:$PATH
```

Note: Here, `$ORACLE_HOME` is the root directory where you have installed the Mobile Server.

3.7.1.3 DISPLAY

Set the *DISPLAY* environment variable to refer to the X Server that will display the installer and Oracle Database Mobile Server. The format of the *DISPLAY* environment variable is:

```
hostname:display_number.screen_number
```

For example, set the *DISPLAY* environment variable, as follows:

```
setenv DISPLAY myhost:0.0
```

Oracle Database Mobile Server requires a running X server to properly create graphics for the installer, Web applications, and management tools. The frame buffer X server installed with your operating system requires that you remain logged in and have the frame buffer running at all times. If you do not want to do this, then you must use a virtual frame buffer, such as X Virtual Frame Buffer (XVFB) or Virtual Network Computing (VNC).

See Also:

- Your operating system documentation for more information on the *DISPLAY* environment variable.
- Oracle Technology Network (<http://www.oracle.com/technetwork/documentation/index.html>) for further information about obtaining and installing XVFB or other virtual frame buffer solutions. Search OTN for "frame buffer".

3.7.1.3.1 Installing From a Remote Machine Setting the *DISPLAY* environment variable enables you to run the Oracle Universal Installer remotely from another workstation. On the system where you launch the Oracle Universal Installer, set *DISPLAY* to the system name or IP address of your local workstation.

Note: You can use a PC X emulator to run the installer if it supports a PseudoColor color model or PseudoColor visual. Set the PC X emulator to use a PseudoColor visual, and then start the installer. Refer to the X emulator documentation for instructions on how to change the color model or visual settings.

If you get an Xlib error similar to "Failed to connect to server", "Connection refused by server," or "Can't open display" when starting the installer, then run the commands on your local workstations as listed in the following table.

Shell Types	On the Server Host Machine Where the Installer is Running	In the Session on Your Host
C Shell	prompt> setenv DISPLAY <hostname>:0.0	prompt> xhost +<hostname>
Bourne/Korn Shell	prompt> DISPLAY=<hostname>:0.0;export DISPLAY	prompt> xhost +<hostname>

3.7.1.4 TMP and TMPDIR

During installation, Oracle Universal Installer uses a temporary directory for swap space. This directory must meet the requirements listed in [Section 3.6, "System Requirements For UNIX Systems"](#) before installing Oracle Database Mobile Server.

The installation may fail if you do not have sufficient space. The installer checks for the `TMP` and `TMPDIR` environment variable to locate the temporary directory. If the `TMP` environment variable is not set, then the installer uses the `/tmp` directory. If the `TMPDIR` environment variable is not set, then the installer uses the `/var/tmp` directory. Set the `TMP` and `TMPDIR` environment variable using the commands in the following table.

C Shell	Bourne/Korn Shell
prompt> setenv TMP <i>full_path</i>	prompt> TMP= <i>full_path</i> ;export TMP
prompt> setenv TMPDIR <i>full_path</i>	prompt> TMPDIR= <i>full_path</i> ;export TMPDIR

3.7.2 Pre-Install Requirements Before Installing the MDK on LINUX

Of all of the UNIX platforms, the MDK is only supported on the Linux platform.

But before installing the MDK on your Linux system, set the following environment variables:

Table 3–4 Linux Environment Variable

Linux Environment Variable	Description
JDK_HOME	Point to the JDK installation directory
ORACLE_HOME	ORACLE_HOME point to the root folder of the Mobile Development Kit installation

3.7.3 Creating UNIX Accounts and Groups

The following UNIX account and groups are required for the installation process:

- [Section 3.7.3.1, "UNIX Group Name For the Oracle Universal Installer Inventory"](#)
- [Section 3.7.3.2, "UNIX Account to Own Oracle Software"](#)

3.7.3.1 UNIX Group Name For the Oracle Universal Installer Inventory

Use the `admintool` or `groupadd` utility to create a group name. In the following text the group name is `oinstall`. The `oinstall` group will own Oracle Universal Installer's `oraInventory` directory. The `oracle` user account that runs the installer must have the `oinstall` group as its primary group and `dba` as its secondary group.

For more information on these utilities, refer to your operating system documentation.

3.7.3.2 UNIX Account to Own Oracle Software

The `oracle` account is the UNIX account that owns Oracle software for your system. You must run Oracle Universal Installer from this account.

Create an `oracle` account with the properties listed in [Table 3–5](#).

Table 3–5 Oracle Account Properties

Variable	Property
Login Name	Select any name to access the account. This document refers to the name as the <code>oracle</code> account.
Group Identifier	The <code>oinstall</code> group.
Home Directory	Select a home directory consistent with other user home directories.

Table 3–5 (Cont.) Oracle Account Properties

Variable	Property
Login Shell	The default shell can be either the C, Bourne, or Korn shell.

Note: Do not use root as the oracle account.

3.7.4 Configuring Kernel Parameters and Shell Limits for UNIX

Depending on your operating system, see one of the following sections for information on checking the software requirements:

- [Section 3.7.4.1, "Configuring Shell Limits and System Configuration Parameters on AIX"](#)
- [Section 3.7.4.2, "Configuring Kernel Parameters on HP-UX"](#)
- [Section 3.7.4.3, "Configuring the Kernel Parameters on Linux"](#)
- [Section 3.7.4.4, "Set Shell Limits for the User oracle"](#)

3.7.4.1 Configuring Shell Limits and System Configuration Parameters on AIX

On AIX systems, you do not need to configure kernel parameters. However, Oracle recommends that you set shell limits and system configuration parameters as described in this section.

3.7.4.1.1 Configuring Shell Limits for AIX Systems Verify that the shell limits shown in the following table are set to the values shown. The procedure following the table describes how to verify and set the values.

Table 3–6 Shell Limits

Shell Limits as shown in smit	Recommended Value
Soft FILE size	-1 (Unlimited)
Soft CPU time	-1 (Unlimited) -- this is the default value
Soft DATA segment	-1 (Unlimited)
Soft STACK size	-1 (Unlimited)

To view or change the current value specified for these shell limits, follow these steps:

1. Enter the smit command: `#smit chuser`
2. In the "User Name" field, enter the user name of the Oracle software owner, such as `oracle`.
3. Scroll down the list and verify that the value shown for the soft limits listed in the previous table is -1. If necessary, modify the existing value to be -1.
4. Press F10 to exit.

3.7.4.1.2 Configure System Configuration Parameters for AIX Verify that the "Maximum Number of Processes" allowed for each user is set to 2048 or greater. The procedure following the table describes how to verify and set the value.

1. Enter the smit command: `#smit chgsys`

2. Verify that the value shown for "Maximum Number of Processes" allowed per user is greater than or equal to 2048.
3. Press F10 to exit.

Ensure that the ARG_MAX setting is set to the maximum value for AIX 5L:

1. Check the ARG_MAX setting, as follows:

```
prompt> getconf ARG_MAX
```

2. If the value is less than 524288, then run the following command as the root user:

```
#chdev -l sys0 -a ncargs=128
```

3.7.4.2 Configuring Kernel Parameters on HP-UX

Verify that the kernel parameters shown in the following table are set either to the formula shown or to values greater than or equal to the recommended value shown. The procedure following the table describes how to verify and set the values.

Table 3-7 Kernel Parameters for HP-UX

Parameter	Recommended Formula or Value
nfile	3000
nproc	2048

To view or modify the current value or formula specified for these kernel parameters, do the following:

1. Optionally, set the DISPLAY environment variable to specify the display of the local system, as follows:
 - Bourne, Bash, or Korn shell:


```
$ DISPLAY=localhost:0.0 ; export DISPLAY
```
 - C shell:


```
$ setenv DISPLAY localhost:0.0
```
2. Start System Administration Manager (SAM): `#!/usr/sbin/sam`
3. Choose the "Kernel Configuration" area, then choose the "Configurable Parameters" area.
4. Check and possibly modify the value or formula specified for each of these parameters.
5. Exit from SAM.
6. If you modified the value specified for any parameter, then reboot the system with the following: `#!/sbin/shutdown -r -now`
7. If necessary, when the system restarts, log in and switch the user to `root`.

3.7.4.3 Configuring the Kernel Parameters on Linux

Verify that the kernel parameters shown in the following table are set either to the formula shown, or to values greater than or equal to the recommended value shown. The procedures following the table describe how to verify and set the values.

Table 3–8 Kernel Parameters on Linux

Parameter	Value	File
file-max	131072	/proc/sys/fs/file-max

To view or modify the current value specified for these kernel parameters, do the following:

1. Enter the following command to view the current value of the `file-max` kernel parameter:

```
# /sbin/sysctl -a | grep file-max
```

2. To modify the value, do the following:

- a. Create or edit the `/etc/sysctl.conf` file and add the following.

```
fs.file-max = 131072
```

By specifying the values in the `/etc/sysctl.conf` file, they persist when you reboot the system.

- b. Change the current values of the kernel parameter with the following command:

```
# /sbin/sysctl -p
```

Review the output from this command to verify that the values are correct. If the values are incorrect, then perform these steps again.

- c. On UnitedLinux only, enter the following command to cause the system to read the `/etc/sysctl.conf` file when it reboots:

```
# chkconfig boot.sysctl on
```

3.7.4.4 Set Shell Limits for the User oracle

To improve the performance of the software on Linux systems, you must increase the following shell limits for the oracle user, depending on the user's default shell:

Table 3–9 Shell Limits for Linux Systems

Bourne or Bash Shell Limit	Korn Shell Limit	C or tcsh Shell Limit	Hard Limit
nofile	nofile	descriptors	16384
nproc	processes	maxproc	16384

To increase the shell limits, do the following:

1. Add the following lines to the `/etc/security/limits.conf` file, where the arrow (`->`) represents the tab character:

```
* -> -> soft -> nproc -> -> 2047
* -> -> hard -> nproc -> -> 16384
* -> -> soft -> nofile -> -> 2047
* -> -> hard -> nofile -> -> 16384
```

2. Add the following line to the `/etc/pam.d/login` file, if it does not already exist:

```
session required /lib/security/pam_limits.so
```

3. Depending on the oracle user's default shell, make the following changes to the default shell start-up file:

- For the Bourne, Bash, or Korn shell, add the following lines to the `/etc/profile` file:

```
if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
        ulimit -p 16384
        ulimit -n 16384
    else
        ulimit -u 16384 -n 16384
    fi
fi
```

- For the C or tcsh shell, add the following lines to the `/etc/csh.login` file:

```
if ( $USER == "oracle" ) then
    limit maxproc 16384
    limit descriptors 16384
endif
```

3.8 System Requirements for Mobile Clients

The system requirements for mobile client are covered in the *Oracle Database Mobile Server Mobile Client Guide*.

3.9 Setting Up Location of the Datafile on the Server

If you do not want to have the datafile for your Oracle Database Mobile Server applications stored in the default location in the Oracle database, then modify the database configuration file to include the directory where you want your datafile stored. Configure the default directory for new data files in the `db_create_file_dest` parameter. Once updated, restart the Oracle database. This must be done before installing Oracle Database Mobile Server. Refer to your database administration guide for information on how to modify the `db_create_file_dest` parameter.

3.10 Defining Synchronization Tablespace

By default, the synchronization tablespace is SYNCSEVER, and is stored in the `mobilexx.dbf` file in the default location for the database under `<ORACLE_HOME>`, where `xx` is a number between 1 and 25. Here `<ORACLE_HOME>` is the root installation directory of Oracle Database. The tablespace name, filename, and file location for the tablespace is defined in the `<ORACLE_HOME>/Mobile/Server/admin/consolidator_o8a.sql` script file, which is executed during the Mobile Server repository installation process. Here `<ORACLE_HOME>` is the root installation directory of Database Mobile Server. So, if you want to modify the tablespace, there are a few tasks you need to perform BEFORE you install the mobile server.

Tablespace layout across multiple disks can improve the performance of mobile server data synchronization, as it reduces movement of the disk heads and improves I/O response time.

For more information on how to alter the synchronization tablespace, see Section 1.2.6, "Synchronization Tablespace Layout" in the *Oracle Database Mobile Server Troubleshooting and Tuning Guide*.

Installation of Oracle Database Mobile Server

This chapter describes all of the details for you to install Oracle Database Mobile Server, including the following:

- [Section 4.1, "Oracle Database Mobile Server Installation Considerations"](#)
- [Section 4.2, "Starting Oracle Universal Installer"](#)
- [Section 4.3, "Installing Oracle Database Mobile Server"](#)
- [Section 4.4, "Starting and Stopping Mobile Server"](#)
- [Section 4.5, "Testing Your Mobile Server Installation"](#)
- [Section 4.6, "Removing Demo Applications"](#)
- [Section 4.7, "How to Uninstall Oracle Database Mobile Server"](#)

4.1 Oracle Database Mobile Server Installation Considerations

Note the following issues before you start your installation:

- [Section 4.1.1, "Installing Multiple Languages on a Single Solaris Machine"](#)
- [Section 4.1.2, "National Language Support for Chinese, Japanese, and Korean"](#)
- [Section 4.1.3, "Mobile Server on a DHCP Server is Not Supported"](#)
- [Section 4.1.4, "Installing Mobile Server Using Multiple Languages"](#)

4.1.1 Installing Multiple Languages on a Single Solaris Machine

With Oracle Database Mobile Server installed in a Solaris environment, you cannot install multiple languages on a single Solaris machine. Instead, you must perform a separate installation for each language.

4.1.2 National Language Support for Chinese, Japanese, and Korean

The mobile server and Mobile Development Kit have full National Language Support for Simplified Chinese, Japanese, and Korean.

4.1.3 Mobile Server on a DHCP Server is Not Supported

The Oracle Database Mobile Server can only be installed on a server with a static IP address.

4.1.4 Installing Mobile Server Using Multiple Languages

The Oracle Database Mobile Server can be installed with different character sets. It is important to ensure that the back-end Oracle Database supports the character sets as the Mobile Server. Failure to do this will result in unreadable content in the Mobile Manager.

4.2 Starting Oracle Universal Installer

Oracle Database Mobile Server uses Oracle Universal Installer to guide you through each step of the installation process. The Oracle Universal Installer provides the following features:

- Describes installation options for Oracle Database Mobile Server
- Detects pre-set environment variables and configuration settings
- Sets environment variables and configuration during installation
- Offers configuration options for a customized installation of Oracle Database Mobile Server
- Deinstalls products

The Oracle Universal Installer automatically checks your computer prior to installation to verify that your system meets operational requirements. [Table 4-1](#) lists the prerequisite checks that are performed.

Table 4-1 Oracle Universal Installer Automatic Prerequisite Checks

Prerequisite Checks	See Also
Check for enough disk space for installation	Section 3.7, "Recommended System Configuration For UNIX"
On UNIX systems, checks for TMP and TMPDIR variable and sufficient swap space	Section 3.7, "Recommended System Configuration For UNIX"

You start the Oracle Universal Installer using different methods for each type of operating system, as detailed in the following steps:

- [Section 4.2.1, "Starting the Oracle Universal Installer on Windows"](#)
- [Section 4.2.2, "Starting Oracle Universal Installer on UNIX"](#)

4.2.1 Starting the Oracle Universal Installer on Windows

Follow the steps below:

- Log on as a member of the Administrators group for the computer on which you want to install Oracle Database Mobile Server.
- Download the installer package for Oracle Database Mobile Server (DMS) from Oracle technology network and unpack it to the target platform where you want to install the DMS or insert the CD-ROM labeled Oracle Database Mobile Server.
- Execute `Disk1/install/win32/setup.exe` for Windows 32bit system or `Disk1/install/win64/setup.exe` for Windows 64bit system.
- The "Welcome" screen appears.

4.2.2 Starting Oracle Universal Installer on UNIX

Depending on how you got the installer package for Oracle Database Mobile Server, follow below steps to install Database Mobile Server:

Note: Ensure you are **not** logged in as the root user when you start the Oracle Universal Installer.

1. If you have purchased a CD labeled Oracle Database Mobile Server, insert the CD into the CD-ROM drive. Mount the installation CD-ROM. Run Oracle Universal Installer from the CD-ROM:

- a. Log in as the oracle user.

- b. Start the installer by entering the following commands:

```
On Linux 32bit system, prompt> mount_point/Disk1/install/linux/runInstaller
On Linux 64bit system, prompt> mount_
point/Disk1/install/linux64/runInstaller
On Solaris SPARC system, prompt> mount_
point/Disk1/install/solaris/runInstaller
On Solaris Intel system, prompt> mount_
point/Disk1/install/intelsolaris/runInstaller
On AIX system, prompt> mount_point/Disk1/install/aix/runInstaller
On HP-UX Itanium system, prompt> mount_
point/Disk1/install/hpia64/runInstaller
```

2. If you have downloaded the installer package for Oracle Database Mobile Server from Oracle technology network, unpack the installer package to the target platform:

```
On Linux 32bit system, execute Disk1/install/linux/runInstaller
On Linux 64bit system, execute Disk1/install/linux64/runInstaller
On Solaris SPARC system, execute Disk1/install/solaris/runInstaller
On Solaris Intel system, execute Disk1/install/intelsolaris/runInstaller
On AIX system, execute Disk1/install/aix/runInstaller
On HP-UX Itanium system, execute Disk1/install/hpia64/runInstaller
```

This launches Oracle Universal Installer, which installs Oracle Database Mobile Server.

4.2.2.1 Using the oraInventory Directory

The Oracle Universal Installer creates the `oraInventory` directory the first time it is run on a computer. The `oraInventory` directory keeps an inventory of products that the Oracle Universal Installer installs on your computer, as well as other installation information. If you have previously installed Oracle products, then you may already have an `oraInventory` directory.

When a UNIX group name is created and specified, the Oracle Universal Installer grants the specified group the permission to write to the `oraInventory` directory. If another group attempts to run the installer, then they must have permission to write to the `oraInventory` directory. If they do not have permission, then the installation will fail.

The location of `oraInventory` is defined in the `oraInst.loc` file. See [Table 4-2](#) for the location of the `oraInst.loc` file for your system.

The latest installation log file is stored in:

```
/your_base_directory/oraInventory/logs/installActionstodays_date_time.log
```

The `your_base_directory` identifier is the location for your installation files and `today's_date_time` is the date and time of installation. Log file names of previous installation sessions take the form `installActionstoday's_date_time.log`.

Do not delete or manually alter the `oraInventory` directory or its contents. Doing so can prevent the installer from locating products that you have installed on your system.

4.2.2.2 Location of Files on UNIX

The following table lists the location of the `oratab` and `oraInst.loc` file for each platform:

Table 4–2 File Locations for Each Platform

Platform	oratab and emtab	oralnst.loc
AIX	/etc	/etc
Solaris	/etc	/var/opt/oracle
Linux	/etc	/etc
HP-UX	/etc	/var/opt/oracle

4.3 Installing Oracle Database Mobile Server

Oracle Database Mobile Server consists of two main components: the Oracle Database Mobile Server and Mobile Development Kit. Each is installed separately from the Oracle Database Mobile Server installer package. The following sections describe all your installation options:

- [Section 4.3.1, "GUI Install of the Mobile Server and MDK Components"](#)
- [Section 4.3.2, "Silent Install of the Mobile Server and MDK Components"](#)
- [Section 4.3.3, "Providing High Availability with a Farm of Multiple Mobile Servers"](#)
- [Section 4.3.4, "Install Oracle Database Mobile Server on Linux"](#)
- [Section 4.3.5, "Mobile Client Install"](#)

4.3.1 GUI Install of the Mobile Server and MDK Components

Once the Oracle Universal Installer is initiated, perform the following steps to install Oracle Database Mobile Server:

1. On the "Welcome" screen, click "Next".
2. On the "Installation Types" screen, choose the type of installation to execute. Your options are as follows:
 - Mobile Server
 - Mobile Development Kit

Each of these Install options is discussed in the following sections:

- [Section 4.3.1.1, "Installation of the Mobile Server"](#)
- [Section 4.3.1.2, "Installation of Mobile Development Kit"](#)

4.3.1.1 Installation of the Mobile Server

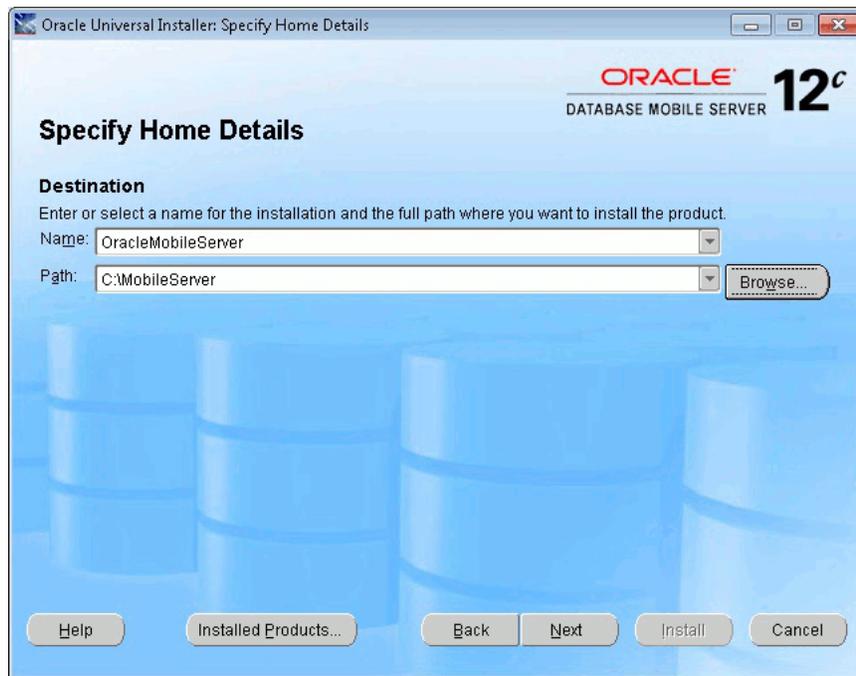
Install the Oracle Database Mobile Server from the Installation Types screen, as follows:

1. Choose "Mobile Server" and click "Next" (see, [Figure 4-1](#)). On Solaris, AIX and HP-UX platforms, only the "Mobile Server" installation type is available, so the "Select Installation Type" screen is not displayed.

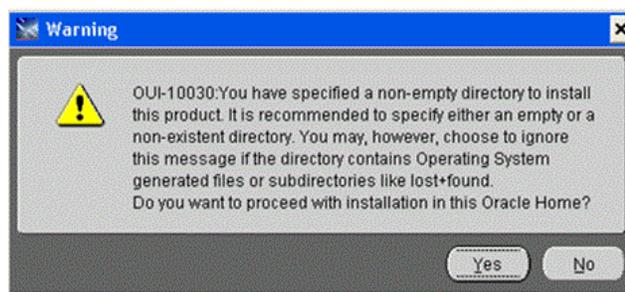
Figure 4-1 Select Mobile Server



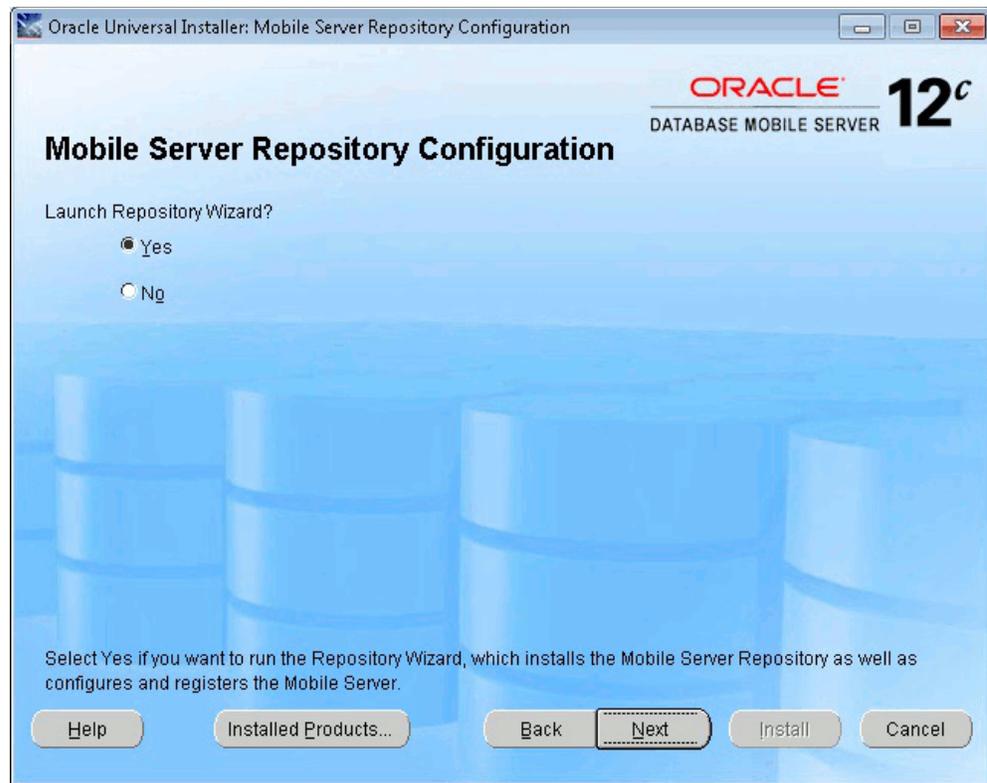
2. On the "Specify Home" screen, as seen in [Figure 4-2](#), enter the name for your mobile server installation and path where you want to install Oracle Database Mobile Server. Both fields cannot include any spaces. The path you specified is known as Oracle Home.

Figure 4–2 Specify Home Details

3. Choose an empty directory to install Mobile Server. If the installation location you specified is not empty, you will get the warning as shown in [Figure 4–3](#), you can ignore this warning.

Figure 4–3 OUI Check

4. Click "Yes" to install the mobile server repository. Click "Yes" to launch Repository Wizard to install and configure mobile server repository during the installation process. Click "No" to postpone the installing and configuring of mobile server repository to after the installation process. Click "Next".

Figure 4-4 Mobile Server Repository Configuration

5. Click "Yes" or "No" to indicate if you want the demo applications installed on the server. Click "Next".

Figure 4-5 Mobile Server Demo Application Configuration

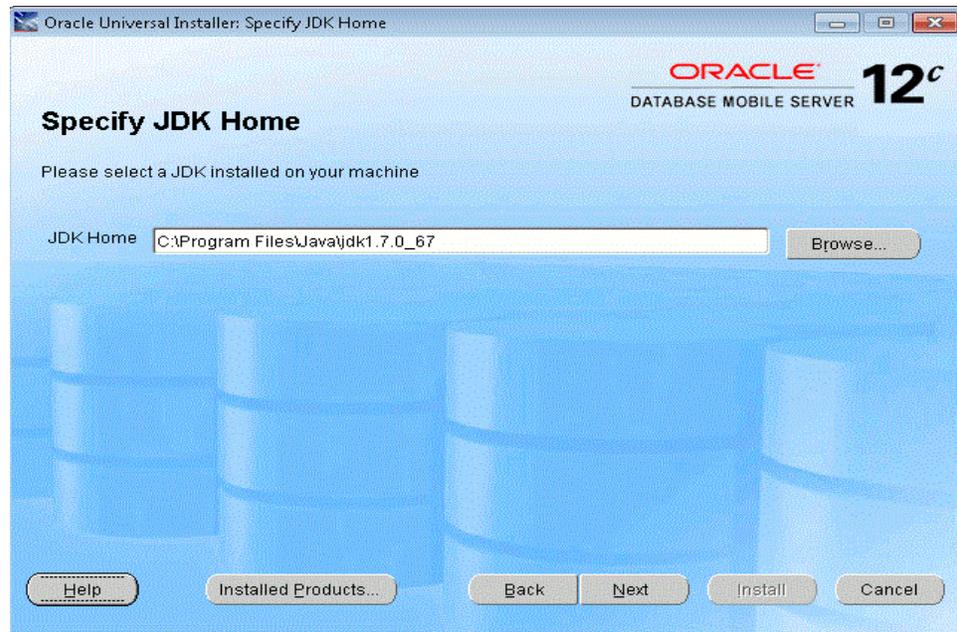
6. Select the installation directory of your application server (on the "Specify Application Server Home" screen).

- If you choose WebLogic as application server, specify WL_HOME as Application Server Home, where WL_HOME is the top-level installation directory for WebLogic Server.
- If you choose Oracle GlassFish server or GlassFish server Open Source Edition as application server, specify as-install as Application Server Home, where as-install is the base installation directory for GlassFish Server.
- If you choose TomEE as application server, specify CATALINA_HOME as Application Server Home, where CATALINA_HOME is the installation directory of TomEE Server.

Figure 4–6 Specify Application Server Home

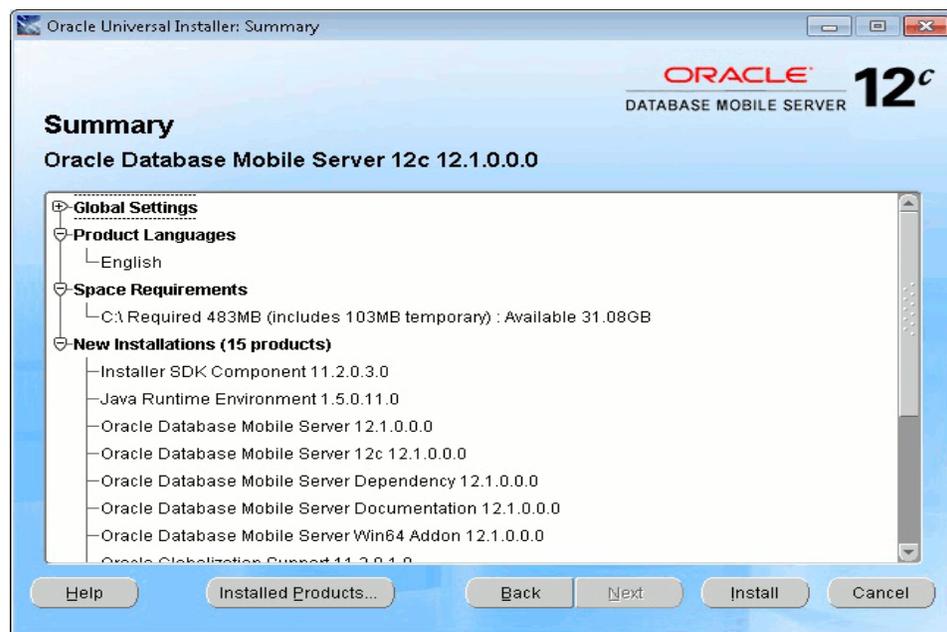


7. Select a JDK installed on your machine (on the "Specify JDK Home" screen). For more information, refer to [Section 3.3.2, "JDK Platform Support"](#). If your application server is WebLogic server, choose the same JDK you use for the WebLogic server. The JDK should be the version as described in [Table 3–1](#), otherwise, Mobile server will use the JDK for your WebLogic server, and not the one you specify in this screen.

Figure 4-7 Specify JDK Home

8. On the "Summary" page, click "Install" to start the installation of mobile server which is installed in your Oracle Home.

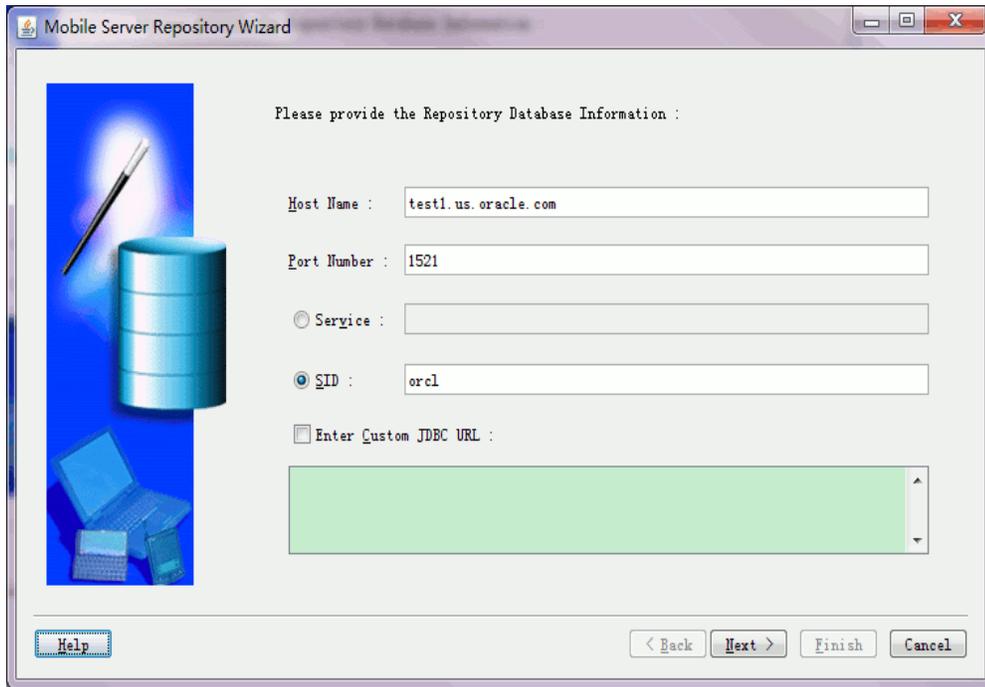
Note: If you previously clicked Yes to install the mobile server repository, then the Repository Wizard is now launched to create and populate the repository with the Mobile Server schema.

Figure 4-8 Installation of Mobile Server

9. Provide the URL for the back-end Oracle database for the mobile server repository, as follows:

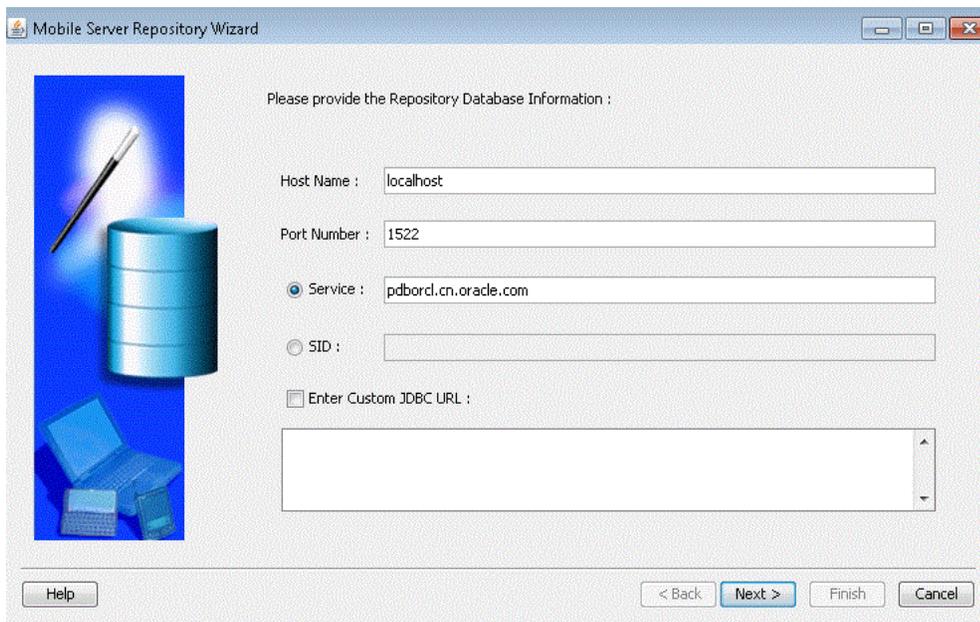
- If you are connecting to a single Oracle database, provide the host name, port and SID or service name. This is shown in [Figure 4-9](#).

Figure 4-9 Provide Repository Database Information



- If you are connecting to a single Oracle 12c Pluggable database (PDB), provide the host name, port number and service name. This is shown in [Figure 4-10](#).

Figure 4-10 Provide Repository Database information with Oracle Database 12c

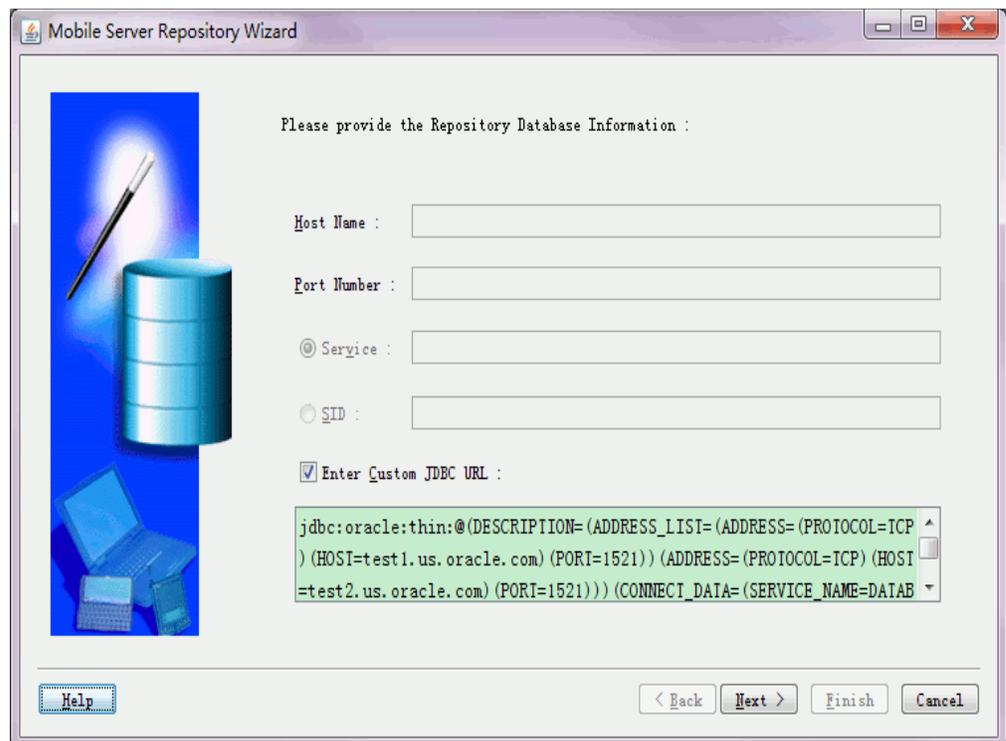


- If you are connecting to an Oracle RAC database, select the "Enter Custom JDBC URL" checkbox and enter the JDBC URL for this database. For an example of using Oracle Database Express Edition, see [Figure 4–11](#).

An Oracle RAC database URL takes the following format:

```
jdbc:oracle:thin:@(DESCRIPTION=
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=TCP)(HOST=PRIMARY_NODE_HOSTNAME)(PORT=1521))
    (ADDRESS=(PROTOCOL=TCP)(HOST=SECONDARY_NODE_HOSTNAME)(PORT=1521))
  )
  (CONNECT_DATA=(SERVICE_NAME=DATABASE_SERVICENAME)))
```

Figure 4–11 Enter JDBC URL for Back-End Database in Repository Wizard



The mobile server installs its meta-data in the back-end database. If you are not sure of the values to enter for the URL, then you can query the data dictionary to obtain these values.

For Oracle 10g and 11g Database, connect as `SYSTEM` user and run the following queries to retrieve the SID:

```
select instance_name from v$instance;
```

For Oracle 12c, connect as the administrator user of your Pluggable database and run the following queries to retrieve the service name:

```
select name from dba_services;
```

If you need to retrieve the port number, check the `listener.ora` file on the machine where the Oracle Database is installed.

Click "Next".

Note: You must enter the user name and password for the following 3 different accounts:

- The Database Administrator account:
 - for Oracle database 10g or 11g, it is a master account to create the mobile server repository schema/user
 - for Oracle database 12c, it is a master account of your pluggable database to create the mobile server repository schema/user in that pluggable database
 - The Mobile Server Repository account: Used to access all the database objects used by the mobile server.
 - The Mobile Server Administrator account: Used to log into the Mobile Manager.
-

10. The Repository Wizard asks for a user name and password for the Database Administrator account, which will be used to connect to the back-end database, create the schema and assign database privileges for the mobile server.

Figure 4–12 Specify Username/Password

Before you provide the user name, this user has to have the following privileges to create the mobile server repository in the back-end database:

- CREATE TABLESPACE, CREATE USER
- The following privileges are required with the Admin option:
 - ALTER ANY TABLE, ALTER SESSION, ALTER SYSTEM, ANALYZE ANY, CREATE SESSION, CREATE ANY SEQUENCE, CREATE ANY VIEW, CREATE ANY TRIGGER,

CREATE ANY INDEX, CREATE ANY TABLE, CREATE ANY SYNONYM, CREATE ANY PROCEDURE, CREATE PROCEDURE, CREATE SEQUENCE, CREATE SYNONYM, CREATE TABLE, CREATE VIEW, CREATE INDEXTYPE, DELETE ANY TABLE, DROP ANY SEQUENCE, DROP ANY PROCEDURE, DROP ANY VIEW, DROP ANY SYNONYM, DROP ANY TRIGGER, DROP ANY INDEX, DROP ANY TABLE, INSERT ANY TABLE, SELECT ANY TABLE, SELECT ANY DICTIONARY, SELECT_CATALOG_ROLE, UPDATE ANY TABLE

Note: You can set the mobile server privileges independently through an API. See Section 3.7, "Set DBA or Operational Privileges for the Mobile Server" in the *Oracle Database Mobile Server Developer's Guide*.

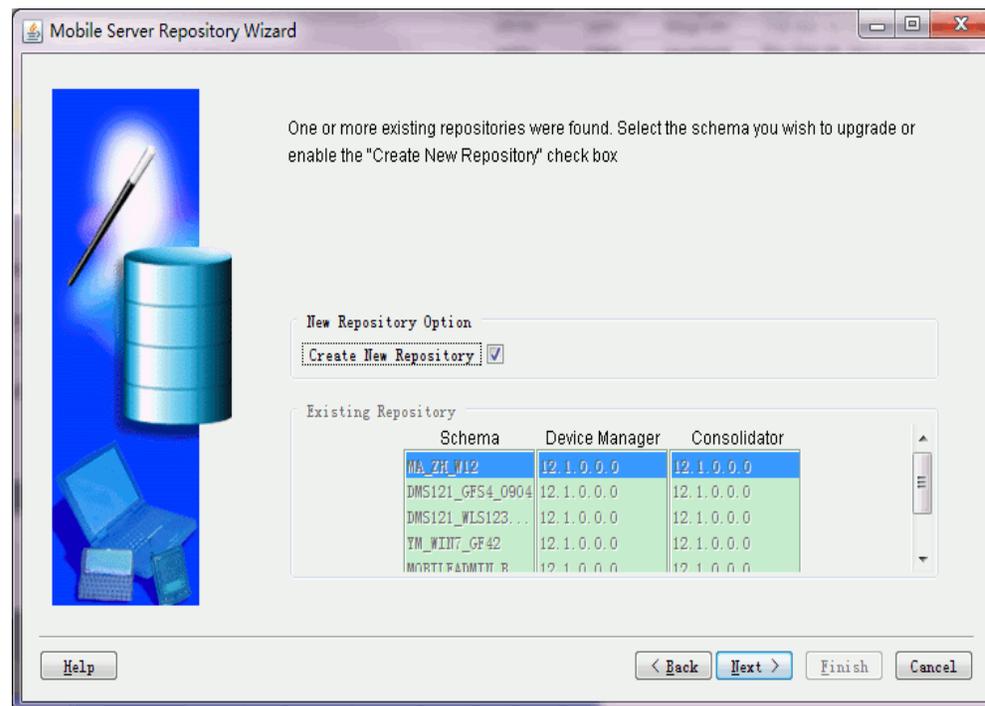
Click "Next".

11. If this is a new repository, the following message is displayed:

No existing repository found. A new repository will be installed.
Otherwise, the following message is displayed (shown in [Figure 4-13](#)):

One or more existing repositories were found.
Select the schema you wish to upgrade or enable the "Create New Repository" check box.

Figure 4-13 Create a New Repository or Upgrade an Existing Repository



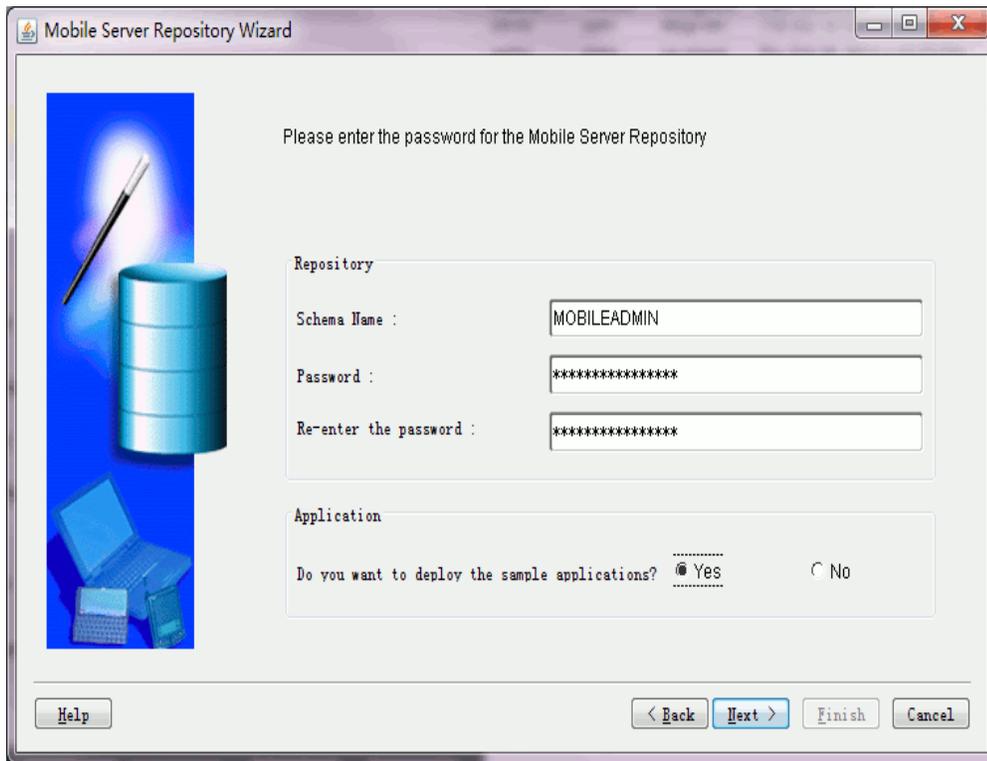
Click "Next".

12. Enter the user name and password for the Mobile Server Repository account. The default name is MOBILEADMIN. This schema contains all database objects used by the mobile server. When the mobile server accesses the repository, it uses the

Mobile Server Repository user name/password. Also, check whether you want to install sample applications or not. Click "Next".

Note: You should install the samples only on a development system; you should NEVER install the samples on a production mobile server. Doing so creates a security risk on your production environment.

Figure 4–14 Enter Password for Mobile Server Repository



13. Enter the user name and password for the Mobile Server Repository account that you will use to log into the Mobile Manager.

Figure 4–15 Mobile Server Administrator

14. Enter the Schema Name and Password for the schema where the demo applications are installed. When the user accesses the user data in the Mobile Server repository, the Mobile Server authenticates the schema user name and password before access is provided to the user data. Enter the Password for the sample users.

Note: If you choose to install the samples, then the following mobile user accounts will be created: S11U1, JOHN, JANE, JACK, and JUNE. The password of the users is the one you specified in "Samples Clients Password".

Figure 4–16 Enter Username and Password for the Demo Schema

15. Complete the required information to deploy the Mobile Server on a WebLogic, GlassFish or TomEE Server.

To deploy the Mobile Server on a WebLogic Server or GlassFish Server in a new domain, follow the instructions below:

- Choose "Yes" for "Create a New Domain for the Mobile Server" option.
- Choose "Yes" if you want to install iOS MDM support for the "Create an HTTPS listener for MDM" option, otherwise choose "No".
- Specify a "Domain Name" for the new domain.
- Select "Administration Server" (in Server Type) if deploying the Mobile Server on a WebLogic Administration Server, or select Managed Server if deploying the Mobile Server on a WebLogic Managed Server. The "Server Type" is only required if you are deploying the Mobile Server on a WebLogic Server.
- Specify a name for the WebLogic Managed Server in "Managed Server Name". The "Managed Server Name" is only required if you are deploying the Mobile Server on a WebLogic Managed Server.
- In Admin Port:
 - Specify an "Administrative Server" port for WebLogic. This is required for both - that is, if you are deploying the Mobile Server on a WebLogic Administration Server or on a Managed Server.
 - Specify an "Administration Port" for GlassFish.
Ensure the port you specified is available.
- In Server Port:
 - Specify a "Managed Server" port for WebLogic. The "Server Port" is only required if you are deploying the Mobile Server on a WebLogic Managed Server.
 - Specify a "Server Port" for GlassFish.
Ensure the port you specified is available.
- Specify a port number for WebLogic or GlassFish HTTPS listener in MDM Port if you specify to install iOS MDM support. This listener will be configured with two-way SSL.
- Specify a name for the domain administrator in Admin Username. If you are deploying the Mobile Server on a WebLogic Server, the administrator name is set to weblogic.
- Specify a password for the domain administrator in Admin Password. The WebLogic Server requires that the password is at least eight characters in length and contains at least one non-alphabetic character; otherwise, you will receive the following error: *"The value for Admin Password is not valid."*

Note: The WebLogic domain directory is created by default as %APP_SERVER_HOME%\..\user_projects\domains\Domain Name%; the GlassFish domain directory is created as %APP_SERVER_HOME%\domains\Domain Name%. For example:

If WebLogic is installed at C:\Oracle\Middleware\wlserver_10.3, and you specify "mobileserver" as the new domain name, then C:\Oracle\Middleware\user_projects\domains\mobileserver will be created and domain configuration files are stored at that directory.

If GlassFish is installed at C:\glassfish3\glassfish, and you specify "mobileserver" as the new domain name, then C:\glassfish3\glassfish\domains\mobileserver will be created and domain configuration files are stored at that directory.

Figure 4-17 Deploy the Mobile Server on a New WebLogic Server Domain

Mobile Server Repository Wizard

Please provide information to deploy Mobile Server on your web application server

Deploy Mobile Server

Create a new domain for Mobile Server? Yes No

Create an HTTPS listener for MDM? Yes No

Please specify properties for a new domain

Domain Name :

Server Type : Administration Server Managed Server

Managed Server Name :

Admin Port :

Server Port :

Admin Username :

Admin Password :

Re-Enter Password :

Help < Back Next > Finish Cancel

Figure 4–18 Deploy the Mobile Server on a New GlassFish Server Domain

Please provide information to deploy Mobile Server on your web application server

Deploy Mobile Server

Create a new domain for Mobile Server? Yes No

Create an HTTPS listener for MDM? Yes No

Please specify properties for a new domain

Domain Name :

Admin Port :

Server Port :

Admin Username :

Admin Password :

Re-Enter Password :

[Help](#)

To deploy the mobile server on an existing domain under WebLogic and Glassfish, follow the instructions:

- Choose "No" for the "Create a new domain for the Mobile Server" option.
- Choose "Yes" if you want to install iOS MDM support for the "Create an HTTPS listener for MDM" option, otherwise choose "No".
- Specify a name for the existing domain in Domain Name. The "Domain Name" is only required if you are deploying the Mobile Server on a GlassFish Server.
- Specify the "Domain Directory" for the existing domain.

Note: For WebLogic, set the domain directory to the folder where your domain configuration files are stored; for GlassFish, set the domain directory to the parent of the folder where your domain configuration files are stored. For example:

If you want to deploy mobile server on an existing WebLogic domain called "msDomain", and the domain configuration files are stored at C:\Oracle\Middleware\user_projects\domains\msDomain, then you should specify "C:\Oracle\Middleware\user_projects\domains\msDomain" as the Domain Directory.

If you want to deploy mobile server on an existing GlassFish domain called "msDomain", and the domain configuration files are stored at C:\glassfish3\glassfish\domains\msDomain, then you should specify "C:\glassfish3\glassfish\domains" as the Domain Directory.

- Select "Administration Server" (in Server Type) if you are deploying the Mobile Server on a WebLogic Administration Server, or select Managed Server

if you are deploying the Mobile Server on a WebLogic Managed Server. The "Server Type" is only required if deploying the Mobile Server on a WebLogic Server.

- Specify the "Administration Port" (in Admin Port) of the existing GlassFish Server domain. The "Admin Port" is only required if you are deploying the Mobile Server on a GlassFish Server.
- Specify the "Server Port" of the existing GlassFish Server domain. The "Server Port" is only required if you are deploying the Mobile Server on a GlassFish Server.
- Specify a port number for WebLogic or GlassFish HTTPS listener in MDM Port if you specify to install iOS MDM support. This listener will be configured with two-way SSL.
- Specify the administrator name of the existing domain (in Admin Username).
- Specify the password of the domain administrator (in Admin Password).

Figure 4–19 Deploy the Mobile Server on an Existing WebLogic Server Domain

Mobile Server Repository Wizard

Please provide information to deploy Mobile Server on your web application server

Deploy Mobile Server

Create a new domain for Mobile Server? Yes No

Create an HTTPS listener for MDM? Yes No

Please specify properties for an existing domain

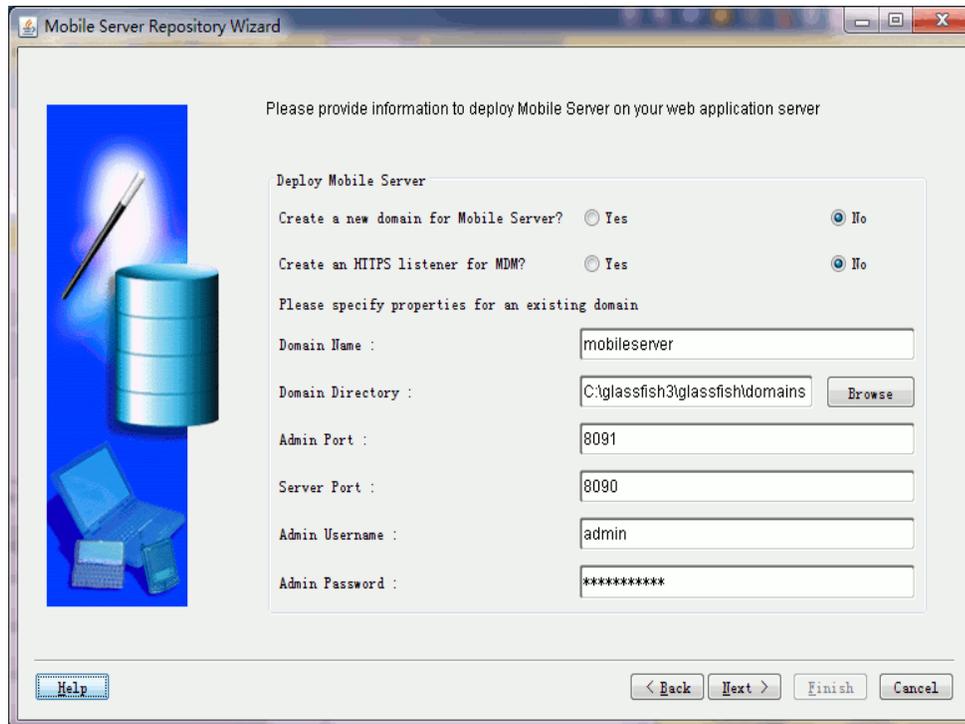
Domain Directory : C:\Oracle\wls1212\user_projects\domains\mobileserver

Server Type : Administration Server Managed Server

Managed Server Name : managedServer

Admin Username : weblogic

Admin Password : *****

Figure 4–20 Deploy the Mobile Server on an Existing GlassFish Server Domain

To deploy the mobile server under TomEE, follow the instructions:

- Specify the "Server Port" value with HTTP connector port value in default Service named "Catalina" of TomEE server.

Ensure the port you specified is available.

Note: You can look for the HTTP connector port value in %CATALINA_HOME%\conf\server.xml, where CATALINA_HOME is the installation directory of TomEE server.

For example, the following is part of a content from %CATALINA_HOME%\conf\server.xml. In this example, the HTTP connector port value in default Service named "Catalina" is 8080.

```
<Service name="Catalina">
  <Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
```

16. A summary screen appears informing you if a repository is installed or not. Click "Next".
17. The installation screen appears. Wait until the install is completed. Click "Next".
18. Once the repository wizard is finished, click "Finish" to leave the wizard.
19. Click "Exit" to complete the installation.
20. Review the installation log files repository.log or populate.log for any errors, which are located in `ORACLE_HOME\mobile\server`. The repository.log file

contains all general installation information on a successful installation, and installation errors on a failed installation. Sometimes, if an unexpected exception occurred and was not caught by OUI, a file named populate.log is created and contains the exception stack.

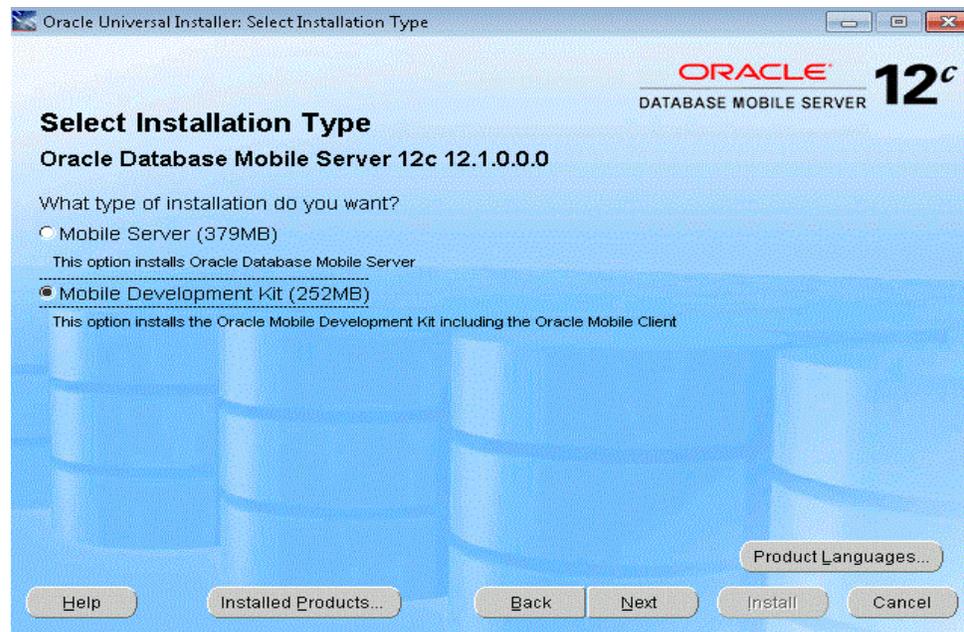
21. When you configure multiple mobile servers against a single repository, this is known as a farm. You must enable the Device Manager, Mobile Manager and mobile client to work properly in a farm environment. For information on configuring all mobile servers to act properly in a farm, see Section 1.2, "Manage Mobile Server Farms" in the *Oracle Database Mobile Server Administration and Deployment Guide*.
22. When installation is completed, launch script runmobileserver on Linux or runmobileserver.bat on Windows to start the Mobile Server.

4.3.1.2 Installation of Mobile Development Kit

Install Mobile Development Kit from the Installation Types screen by performing the following:

1. Choose the "Mobile Development Kit" and clicking "Next".

Figure 4–21 Select Mobile Development Kit



2. On the "Specify Home Details" screen, enter the name and path where you want to install Mobile Development Kit, which cannot include any spaces.

Figure 4–22 Specify Home Details

3. On the "Summary" screen, click Install. Click "Exit" to return to the installation screen.

You have now installed the Mobile Development Kit.

4.3.2 Silent Install of the Mobile Server and MDK Components

You can execute a command-line tool that runs the installer in silent mode. That is, it installs either the Mobile Development Kit or the mobile server for you without having to use the Installation GUI. The following sections describe how to silently install these components on either the Windows or UNIX platforms:

- [Section 4.3.2.1, "Silent Install for Mobile Server or Mobile Development Kit on Windows"](#)
- [Section 4.3.2.2, "Silent Install Mobile Server or the Mobile Development Kit on UNIX"](#)
- [Section 4.3.2.3, "Configuration File for Deploying Mobile Serve"](#)

4.3.2.1 Silent Install for Mobile Server or Mobile Development Kit on Windows

The `setup.exe` command can be executed to run the installer in silent mode for Windows, as follows:

```
setup.exe -responseFile
/full_directory_path/stage/Response/oracle.mobile.install_type_2.rsp
-silent -nowelcome
```

There are two response files that contain variables that define how the mobile server or the MDK are installed. The following sections describe the response files:

- [Section 4.3.2.1.1, "Response File for Mobile Server Install"](#)
- [Section 4.3.2.1.2, "Response File for MDK Install"](#)

4.3.2.1.1 Response File for Mobile Server Install Variables to be filled for silent install in the response file `oracle.mobile.install_type_2.rsp` to install Oracle Database Mobile Server are as follows:

- Define the `ORACLE_HOME` destination directory, as follows:

```
ORACLE_HOME="C:\MobileServer"
```

- Define the name for your Oracle Home, as follows:

```
ORACLE_HOME_NAME="MobileServer"
```

- Provide the hostname, port, and user for the back-end database, where the repository is to be installed, as follows:

```
s_repo_jdbc_url="jdbc:oracle:thin:@<host>:<port>:<sid>"
```

- Specify true or false if the repository is to be created on the back-end database, as follows:

```
b_repository=true
```

- Provide the database system user name and password, as follows:

```
s_db_admin_name="myuser"
s_db_admin_pwd="mypwd"
```

- Provide the mobile server schema name and password, as follows:

```
s_mobile_schema_name="myschema"
s_mobile_schema_pwd="myschemapwd"
```

- Provide the demo schema user name and password, as follows:

```
s_demo_schema_name="demoschema"
s_demo_schema_pwd="demoschemapwd"
```

- Provide the demo client password, as follows:

```
s_demo_client_pwd="Welcome1"
```

- Provide the mobile server administrator user name and password, as follows:

```
s_mobileserver_admin_name="adminuser"
s_mobileserver_admin_pwd="adminpwd"
```

- Specify true or false to indicate whether the samples are to be installed, as follows:

```
b_mobile_samples=true
```

- Specify application server home, as follows:

```
s_appserver_home="/path/to/appserver"
```

Refer to Step 7 in [Section 4.3.1.1, "Installation of the Mobile Server"](#) for how to set application server home.

- Specify the configuration file for mobile server deploy. For more information, refer to [Section 4.3.2.3, "Configuration File for Deploying Mobile Serve"](#). Note that `b_repository` must set to true if you want to deploy mobile server:

```
s_dep_conf_file="/path/to/mobileserver_config.ini"
```

- Specify JDK home, as follows:

```
oracle.mobile.server:java_path="/path/to/jdk"
```

4.3.2.1.2 Response File for MDK Install Variables in the `oracle.mobile.install_type_1.rsp` response file to install MDK, are as follows:

- Define the `ORACLE_HOME` destination directory, as follows:

```
ORACLE_HOME="C:\MDK"
```

- Define the name for your Oracle Home, as follows:

```
ORACLE_HOME_NAME="MDK"
```

- Specify JDK home, as follows:

```
oracle.mobile.windows_mdk:java_path="/path/to/jdk"
```

4.3.2.2 Silent Install Mobile Server or the Mobile Development Kit on UNIX

You can install the mobile server and/or the Mobile Development Kit by using the `runInstaller` command, which is on the top level of the CD or the downloaded ZIP file for the product.

There are two response files that contain variables that define how the mobile server or the MDK are installed, which are located in the `stage/Response` folder of the CD or ZIP file.

4.3.2.2.1 Response File for Mobile Server Install The response file for installing the mobile server is `oracle.mobile.install_type_2.rsp`. The variables that you can modify in this file for defining your install are as follows:

- Define the `ORACLE_HOME` destination directory, as follows:

```
ORACLE_HOME="/home/user/mobileserver"
```

- Define the name for your Oracle Home, as follows:

```
ORACLE_HOME_NAME="MobileServer"
```

- Provide the hostname, port, and user for the back-end database, where the repository is to be installed, as follows:

```
s_repo_jdbc_url="jdbc:oracle:thin:@<host>:<port>:<sid>"
```

- Specify true or false if the repository is to be created on the back-end database, as follows:

```
b_repository=true
```

- Provide the database SYSTEM user name and password, as follows:

```
s_db_admin_name="myuser"
s_db_admin_pwd="mypwd"
```

- Provide the mobile server schema name and password, as follows:

```
s_mobile_schema_name="myschema"
s_mobile_schema_pwd="myschemapwd"
```

- Provide the mobile server demo schema user name and password, as follows:

```
s_demo_schema_name="demoschema"
s_demo_schema_pwd="demoschemapwd"
```

- Provide the demo client password, as follows:

```
s_demo_client_pwd="Welcome1"
```

- Provide the mobile server admin user name and password, as follows:

```
s_mobileserver_admin_name="adminuser"
s_mobileserver_admin_pwd="adminpwd"
```

- Specify true or false to indicate whether the samples are to be installed, as follows:

```
b_mobile_samples=true
```

- Specify application server home, as follows:

```
s_appserver_home="/path/to/appserver"
```

Refer to Step 7 in [Section 4.3.1.1, "Installation of the Mobile Server"](#) for how to set application server home.

- Specify the configuration file for mobile server deploy. Refer to [Section 4.3.2.3, "Configuration File for Deploying Mobile Serve"](#) for more information.

Note: `b_repository` must set to true if you want to deploy mobile server.

```
s_dep_conf_file="/path/to/mobileserver_config.ini"
```

- Specify JDK home, as follows:

```
oracle.mobile.server:java_path="/path/to/jdk"
```

4.3.2.2 Response File for MDK Install The response file for installing the MDK is `oracle.mobile.install_type_1.rsp`. You can define the variables in this file for your install as follows:

- Define the `ORACLE_HOME` destination directory, as follows:

```
ORACLE_HOME="/home/user/mdk"
```

- Define the name for your Oracle Home, as follows:

```
ORACLE_HOME_NAME="MDK"
```

- Specify JDK home, as follows:

```
oracle.mobile.linux_mdk:java_path="/path/to/jdk"
```

4.3.2.3 Execute Silent Install of Mobile Server and MDK on UNIX After modifying the response files, if you want to install both the mobile server and the MDK, then execute the `runInstaller` command twice. Each execution is provided a different response file. The following installs the mobile server, as follows:

```
./runInstaller -responseFile
/full_directory_path_to_response_file/oracle.mobile.install_type_2.rsp
-silent -nowelcome
```

where the options for this command are as follows:

Table 4–3 The runInstaller Command Options

Command Option	Description
responseFile	Provide the path and name of the response file for the component that you are installing. There are two response files: one for installing the mobile server and one for installing the MDK.
silent	Always use this option, as it defines that this is a silent installation.
nowelcome	Stops the GUI from displaying.

4.3.2.3 Configuration File for Deploying Mobile Serve

The configuration file specified in `s_dep_conf_file` variable is used to specify the required information to deploy mobile server. The file must be an INI format file. You can find the sample configuration file at `%ORACLE_HOME%\Mobile\Server\admin\mobileserver_config.ini`.

Refer to [Table 4–4, "Configuration File for Deploying Mobile Server on WebLogic, GlassFish and and TomEE Servers"](#) and [Table 4–5, "Sample Configuration File"](#) for information on specifying the configuration to deploy mobile server:

Table 4–4 Configuration File for Deploying Mobile Server on WebLogic, GlassFish and and TomEE Servers

Parameters	Description
domain_name	String type, required for WebLogic and GlassFish. Name of the WebLogic Server domain or GlassFish Server domain. If deploying the Mobile Server on a WebLogic Server and 'create_new_domain=false', this parameter is not required.
domain_dir	String type, required for WebLogic and GlassFish. Domain directory. Refer to Step 16 in Section 4.3.1.1, "Installation of the Mobile Server" for how to set the domain directory.
admin_user	String type, required for WebLogic and GlassFish. Name of the domain administrator. Note: Use 'weblogic' as the administrator user name if you are creating a new domain for WebLogic Server.
admin_password	String type, required for WebLogic and GlassFish. Password of the domain administrator, please change the password after deployment. Note: WebLogic Server requires that the password is at least eight characters in length and contains at least one non-alphabetic character.
server_port	Integer type, required for GlassFish, TomEE and WebLogic Managed Server. <ul style="list-style-type: none"> ■ Set it to the server port for GlassFish and TomEE. ■ If deploying the Mobile Server on a WebLogic Managed Server and 'create_new_domain=true', this parameter is required, set it to the Managed Server port
admin_port	Integer type, required for WebLogic and GlassFish. <ul style="list-style-type: none"> ■ Set it to domain administration port for GlassFish. ■ If deploying the Mobile Server on either a WebLogic Administration Server or a Managed Server, and 'create_new_domain=true', this parameter is required. Set it to Administration Server port for WebLogic.
create_new_domain	Boolean type, optional. Set to 'true' to create a new domain, or set to 'false' to use an existing domain, default value is 'true'.

Table 4–4 (Cont.) Configuration File for Deploying Mobile Server on WebLogic, GlassFish and TomEE Servers

Parameters	Description
force_deploy	Boolean type, optional. If set to 'true', redeploy the Mobile Server even if it has already been deployed or already exists, default value is 'false'.
managed_server_name	String type, required for WebLogic if deploying the Mobile Server on a WebLogic Managed Server. Managed Server name.
target_server_type	String type, required for WebLogic. <ul style="list-style-type: none"> ▪ Set to 'A' if deploying the Mobile Server on a WebLogic Administration Server. ▪ Set to 'M' if deploying the Mobile Server on a WebLogic Managed Server.

Table 4–5 Sample Configuration File

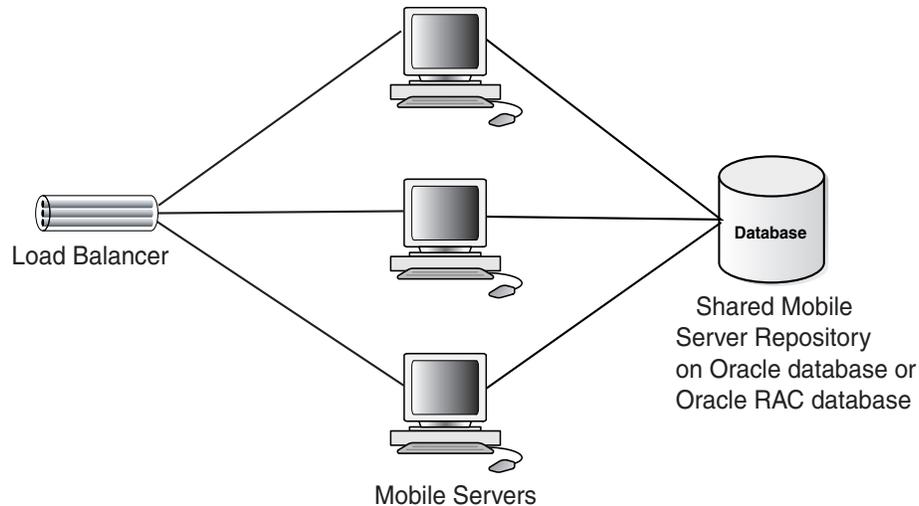
Server Type	Sample Configuration
WebLogic Server configuration	Specify the configuration parameters in [WLS] section [WLS] domain_name=mobileserver_wls domain_dir=C:\Oracle\Middleware\user_projects\domains\mobileserver_wls admin_user=weblogic admin_password=welcome1 admin_port=7001 server_port=7009 managed_server_name=ManagedServer target_server_type=M create_new_domain=true force_deploy=true
GlassFish Server configuration	Specify the configuration parameters in [GFS] section [GFS] domain_name=mobileserver_gfs domain_dir=C:\glassfish3\glassfish\domains admin_user=admin admin_password=welcome1 server_port=8090 admin_port=8089 create_new_domain=true force_deploy=true
TomEE Server configuration	Specify the configuration parameters in [TOMEE] section [TOMEE] server_port=8080 force_deploy=true

4.3.3 Providing High Availability with a Farm of Multiple Mobile Servers

In some cases, you may want to have multiple mobile servers using the same mobile server repository on an Oracle database or an Oracle RAC database. For example, as [Figure 4–23](#) shows, if you wanted to load balance your mobile clients across multiple mobile servers, you could add a load balancer—such as Oracle Web Cache—before the shared mobile servers, and then your clients would be balanced across these mobile servers, of which each accesses the same data in the shared repository.

This is the type of configuration that entails an Oracle Database Mobile Server Farm.

Figure 4–23 Multiple Mobile Servers Sharing a Repository



You can install multiple mobile servers, each on its own host, that use the same mobile server repository, as follows:

1. Install the load balancer. Refer to Section 9.6 "Using a Load Balancer" in *Administration and Deployment Guide* for instructions about how to install a load balancer. Oracle Database Mobile Server certifies Oracle Web Cache as a load balancer. You can use any load balancer as long as the `JSESSIONID` is used.
2. Install the first mobile server and the mobile server repository following the instructions in [Section 4.3.1.1, "Installation of the Mobile Server"](#).
3. Install additional mobile servers. For each subsequent mobile server, follow the instructions in [Section 4.3.1.1, "Installation of the Mobile Server"](#), supplying the same JDBC URL for the back-end Oracle database where the mobile server repository exists. Select Yes for creating the repository, which registers the shared mobile server with the repository.
4. Configure mobile servers with the load balancer URL. Because the mobile client accesses any of these mobile servers through a load balancer, you need to configure each mobile server with the load balancer URL. This ensures that when the `setup.exe` is downloaded by the client, the executable automatically includes the load balancer URL, instead of the mobile server URL.

This process is similar to how the mobile server is configured to use with a reverse proxy. Configure both the `reverse_proxy` and `DM_AUTO_SYNC_CACHE` parameters in the `mobile.ora` configuration file, as follows:

```
[MOBILE]
REVERSE_PROXY=http://<load_balancer_hostname>:<port_number>/mobile
DM_AUTO_SYNC_CACHE=YES
```

5. In a configuration with multiple mobile servers and a load balancer, you must ensure that the load balancer is configured to route HTTP requests within the same HTTP session to the same mobile server. To do this, configure the load balancer to honor the JSESSIONID cookie. When a user logs into a mobile server, an HTTP session is created for that user. An HTTP cookie with the name JSESSIONID is sent back to the mobile client. Subsequent HTTP requests from the mobile client will contain the HTTP cookie, which allows the mobile server to identify the session for that user. Refer to Section 9.6 "Using a Load Balancer" in *Administration and Deployment Guide* for more instructions about how to configure the load balancer to use with a mobile server farm.

If the load balancer uses a different algorithm to distribute HTTP requests, such as round-robin or server-load based, then the mobile server will not work correctly.

6. Perform additional configuration for device management and data synchronization for all mobile servers in the farm as described in Section 1.2, "Manage Mobile Server Farms" in the *Oracle Database Mobile Server Administration and Deployment Guide*.

4.3.4 Install Oracle Database Mobile Server on Linux

The following sections provide considerations when installing Oracle Database Mobile Server on a Linux platform:

- [Section 4.3.4.1, "Patch Required if Using Linux Redhat 3.0"](#)
- [Section 4.3.4.2, "Providing Enough Swap Space on the Linux Platform"](#)

4.3.4.1 Patch Required if Using Linux Redhat 3.0

If you install Oracle Database Mobile Server on Linux Redhat 3.0 and the following error occurs, apply the 3006854 patch and start the installation again.

Error occurred during initialization of VM Unable to load native library:
 /tmp/myhost/jre/lib/i386/libjava.so: symbol __libc_wait, version GLIBC_2.0
 not defined in file libc.so.6 with link time reference.

4.3.4.2 Providing Enough Swap Space on the Linux Platform

If the swap space is not sufficient on your Linux machine, then modify the value for the SWAP_SPACE variable in the `install/linux/oraparam.ini` file. The default value is set to `SWAP_SPACE=1536`. For example, if your machine is not as powerful, decrease the swap space; for example, `SWAP_SPACE=1024`.

4.3.5 Mobile Client Install

The mobile client can use either Berkeley DB, SQLite or JavaDB database. For instructions on how to install the mobile client, see Chapter 2, "Installing the Mobile Client" in the *Oracle Database Mobile Server Mobile Client Guide*.

4.4 Starting and Stopping Mobile Server

The following describes how to start and stop the mobile server on the windows machine and on the UNIX machine:

- [Section 4.4.1, "Starting Mobile Server"](#)
- [Section 4.4.2, "Stopping Mobile Server"](#)

4.4.1 Starting Mobile Server

- To start the Mobile Server on a Windows machine, use one of the options below:
 - On the Windows Start Menu, navigate to *All Programs -> Oracle Database Mobile Server 12c -> Mobile Server*. Click on *Mobile Server* item to start the Mobile Server (Right-click and use *Run as Administrator* on Windows 7 or later version).
 - Open a command prompt (use *Run as Administrator* on Windows 7 or later version), change the directory to %MOBILE_HOME%\Mobile\Server\bin, and run the script runmobileserver.bat.
 - Start the Application Server on which the Mobile Server is deployed. The Mobile Server then automatically starts.
- To start the Mobile Server on a UNIX machine, use one of the options below:
 - Open a shell, change the directory to \$MOBILE_HOME/mobile/server/bin and run the script runmobileserver.
 - Start the Application Server on which the Mobile Server is deployed. The Mobile Server then automatically starts.

4.4.2 Stopping Mobile Server

- To stop the Mobile Server on a Windows machine, use one of the options below:
 - Open a command prompt (use *Run as Administrator* on Windows 7 or later version), change the directory to %MOBILE_HOME%\Mobile\Server\bin, and run the script stopmobileserver.bat.
 - Stop the Application Server on which the Mobile Server is deployed. The Mobile Server then automatically stops.
- To stop the Mobile Server on a UNIX machine, use one of the options below:
 - Open a shell, change the directory to \$MOBILE_HOME/mobile/server/bin and run the script stopmobileserver.
 - Stop the Application Server on which the Mobile Server is deployed. The Mobile Server then automatically stops.

Note:

- The `runmobileserver` and `stopmobileserver` scripts contain the default Java flags. If you want to modify the flags for how Mobile Server is started or stopped, you can modify the scripts as appropriate.
- On WebLogic Server release 12.1.2, if the Derby Server is installed, it is automatically started when WebLogic Server starts. If you do not want to start the Derby Server, pass `noderby` as an argument to the `startWebLogic` command. For example, in the `runmobileserver.bat` script, call `%WLS_DOMAIN_DIR%\bin\startWebLogic.cmd noderby`.
- If the Mobile Server is deployed on a WebLogic Managed Server, and you are starting the Mobile Server for the first time, ensure the Administration Server is running in the same domain. To start the Administration Server, call the script:
 - `%WLS_DOMAIN_DIR%\bin\startWebLogic.cmd` (Windows)
 - `$WLS_DOMAIN_DIR/bin/startWebLogic.sh` (UNIX)

The `WLS_DOMAIN_DIR` is the WebLogic Server domain directory. For more information, see <http://docs.oracle.com/middleware/1212/wls/START/overview.htm#i1070184> (*Starting an Administration Server with a Startup Script*)

4.5 Testing Your Mobile Server Installation

To verify your installation, open your browser, input Mobile Server URL `http://<Mobile_Server_hostname>:<port>/mobile`, and check that the Mobile Manager login page is displayed, as shown in [Figure 4-24](#).

Figure 4-24 Mobile Manager Login Page



Log on to the "Mobile Manager" with the Mobile Server Administrator user name and password and the Mobile Server Farm Page is displayed.

4.6 Removing Demo Applications

Before you use Oracle Database Mobile Server in a production environment, you may wish to remove the demo applications. To remove the demo applications, execute the following batch file:

```
demoinstaller [-install | -uninstall] <sys_user> <sys_pwd> <mobile_user> <mobile_pwd>
```

where the options are as follows:

Option	Description
-install or -uninstall	-install: install the demos -uninstall: remove the demos
<sys_user> <sys_pwd>	Provide the system user name and password that are used to drop the schema in which the demos are installed.
<mobile_user> <mobile_pwd>	Provide the mobile server log on user name and password, which is necessary to remove the demos from the repository.

The removal process enables you to delete the schema into which the applications were installed. If you did not use the schema for any other purpose, then allow the batch file to remove the demo application schema.

4.7 How to Uninstall Oracle Database Mobile Server

Uninstall the Oracle Database Mobile Server components through the Oracle Universal Installer, also perform the following tasks:

1. Delete the following directories:
 - `ORACLE_HOME\Mobile`
 - If you are using WebLogic, delete the `<app_server_deployment_dir>\mobile` directory.
2. Drop `TABLESPACE SYNCSERVER INCLUDING CONTENTS`
3. Delete the `Mobile01.dbf` database file from the operating system.

Upgrade Oracle Database Mobile Server

The following sections describe the steps for upgrading your Oracle Database Mobile Server software:

- [Section 5.1, "Upgrade Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11g"](#)
- [Section 5.2, "Upgrade Oracle Database Mobile Server 11.1.0 or 11.2.0 to 11.3.0"](#)
- [Section 5.3, "Upgrade Oracle Database Mobile Server 11.1.0 to 11.2.0"](#)
- [Section 5.4, "Upgrade Oracle Database Lite from 10g Release 1 or 2 to 10g Release 3"](#)
- [Section 5.5, "Upgrade Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 12c"](#)
- [Section 5.6, "Upgrade Oracle Database Mobile Server 11g to 12.1.0"](#)

5.1 Upgrade Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11g

Oracle Database Mobile Server 11g contains most of the components of the previous Oracle Database Lite product, with a few exceptions. The Oracle Lite client is no longer supported. Instead, all synchronization originates from either a Berkeley DB or SQLite client. Thus, if you have been using Oracle Database Lite in the past, all Oracle Database Lite Mobile Clients will need to be migrated to either of these supported mobile clients. In addition, your Mobile Server needs to be upgraded to the 11g version of the mobile server. You can either upgrade your Mobile Server from 10.3.0.3.0 to 11.1.0, 11.2.0 and 11.3.0 gradually or from 10.3.0.3.0 to 11.3.0 directly.

The following sections detail how you can upgrade Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11g for both your existing mobile server and mobile client.

- [Section 5.1.1, "Pre-Upgrade Instructions"](#)
- [Section 5.1.2, "Upgrade Mobile Server"](#)
- [Section 5.1.3, "Mobile Server Farm Upgrade"](#)
- [Section 5.1.4, "Upgrade Mobile Client"](#)

Note: "Oracle Database Mobile Server 11g" mentioned in this chapter refers to Oracle Database Mobile Server 11.1.0, 11.2.0 and 11.3.0.

5.1.1 Pre-Upgrade Instructions

You can only upgrade an Oracle Database Lite version 10.3.0.3.0 to Oracle Database Mobile Server 11g. Therefore, if your current version of Oracle Database Lite is previous to version 10.3.0.3.0, this must first be upgraded to Oracle Database Lite 10.3.0.3.0 and all patches applied before upgrading to Oracle Database Mobile Server 11g. Details for upgrading Oracle Database Lite versions 10.1 and 10.2 to Oracle Database Lite version 10.3 is described in [Section 5.4, "Upgrade Oracle Database Lite from 10g Release 1 or 2 to 10g Release 3"](#).

5.1.2 Upgrade Mobile Server

Upgrading Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11g can be performed on an installation in the same ORACLE_HOME or in a separate ORACLE_HOME. [Table 5–1](#) shows what version of Oracle Database Lite 10.3.0.3.0 and application server installation can be upgraded to Oracle Database Mobile Server 11.1.0.

Table 5–1 Upgrading Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11.1.0

From Oracle Database Lite 10.3.0.3.0	To Oracle Database Mobile Server 11.1.0 Base	Same Oracle Home	Separate Oracle Home
Oracle Database Lite 10.3.0.3.0 Standalone	Oracle WebLogic Server 11g Release 1	Not Supported	Supported
	GlassFish 3.1	Not Supported	Supported
	Oracle iAS 10.1.3.5.0	Not Supported	Supported
Oracle Database Lite 10.3.0.3.0 with WebLogic 11g Release 1	Oracle WebLogic Server 11g Release 1	Supported	Supported
	GlassFish 3.1	Not Supported	Supported
	Oracle iAS 10.1.3.5.0	Not Supported	Supported
Oracle Database Lite 10.3.0.3.0 with Oracle iAS 10.1.3.5.0	Oracle WebLogic Server 11g Release 1	Not Supported	Supported
	GlassFish 3.1	Not Supported	Supported
	Oracle iAS 10.1.3.5.0	Supported	Supported

[Table 5–2](#) shows what version of Oracle Database Lite 10.3.0.3.0 and application server installation can be upgraded to Oracle Database Mobile Server 11.2.0 or 11.3.0

Table 5–2 Upgrading Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11.2.0 or 11.3.0

From Oracle Database Lite 10.3.0.3.0	To Oracle Database Mobile Server 11.2.0 or 11.3.0 Base	Same Oracle Home	Separate Oracle Home
Oracle Database Lite 10.3.0.3.0 Standalone	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c	Supported	Supported
	GlassFish 3.1	Supported	Supported
	Oracle iAS 10.1.3.5.0	Supported	Supported
	Apache TomEE 1.0	Supported	Supported
Oracle Database Lite 10.3.0.3.0 with WebLogic 11g Release 1	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c	Supported	Supported
	GlassFish 3.1	Supported	Supported

Table 5–2 (Cont.) Upgrading Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11.2.0 or

From Oracle Database Lite 10.3.0.3.0	To Oracle Database Mobile Server 11.2.0 or 11.3.0 Base	Same Oracle Home	Separate Oracle Home
	Oracle iAS 10.1.3.5.0	Supported	Supported
	Apache TomEE 1.0	Supported	Supported
Oracle Database Lite 10.3.0.3.0 with Oracle iAS 10.1.3.5.0	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c	Supported	Supported
	GlassFish 3.1	Supported	Supported
	Oracle iAS 10.1.3.5.0	Supported	Supported
	Apache TomEE 1.0	Supported	Supported

Upgrade OID

If you use OID 10.1.2 to manage user information for Oracle Database Lite 10.3.0.3.0, upgrade to OID version 10.1.4 or 11.1.1.5.

Note: If you are not currently using OID, but would like to use OID with Oracle Database Mobile Server 11g, follow instructions in [Section 5.1.2.5, "Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory \(OID\)"](#).

[Table 5–3](#) shows supported OID versions when upgrading from Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11g. Refer to *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management 11g Release 1 (11.1.1)* for detailed instructions on how to upgrade OID.

Table 5–3 Upgrading OID

Installed Application Server and OID Version	Upgrade To OID Version
Oracle iAS 10.1.3.5.0 with OID 10.1.2	Oracle iAS 10.1.3.5.0 with OID 10.1.4 or OID 11.1.1.5
Oracle iAS 10.1.3.5.0 with OID 10.1.4	You can continue using Oracle iAS 10.1.3.5.0 with OID 10.1.4 or upgrade to OID 11.1.1.5
WebLogic 11g Release 1 with OID 10.1.4	You can continue using WebLogic 11g Release 1 with OID 10.1.4 or upgrade to OID 11.1.1.5

General Guideline

1. If you are using Oracle Database Lite 10.3.0.3.0 Mobile Server Standalone, migrate to one of the supported application servers on a separate ORACLE_HOME: Apache TomEE, OracleAS, WebLogic, or GlassFish. See [Section 3.3.3, "Certified Application Server Configurations"](#) for details on supported application servers.
2. If you are currently using WebLogic or OracleAS, you can do one of the following:
 - You can continue to use the installed application server as your application server as long as the version is a supported version.

- If you plan to upgrade Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11.1.0, and switch to a different application server, you must install Mobile Server in a separate ORACLE_HOME.
 - If you plan to upgrade Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11.2.0 or 11.3.0, and switch to a different application server, you can install Mobile Server either in the same ORACLE_HOME or in a separate ORACLE_HOME. See [Section 5.2.1.1, "Upgrading in the Same ORACLE_HOME"](#) and [Section 5.2.1.2, "Upgrading in a Separate ORACLE_HOME"](#) for additional information.
3. If you have OracleAS installed as the application server, but you want to migrate to WebLogic Server, refer to Oracle WebLogic SmartUpgrade directions on the following site:
<http://www.oracle.com/technetwork/middleware/downloads/smartupgrade-085160.html>
You must apply all patches before upgrading from Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11g.

5.1.2.1 Upgrade in the same ORACLE_HOME

Refer to [Table 5-1](#) and [Table 5-2](#) for the supported cases for upgrading Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11g in the same ORACLE_HOME.

1. Install Oracle Database Mobile Server 11g in ORACLE_HOME.
 - Launch Oracle Universal Installer. On "Select Installation Type" page, select "Mobile Server".
 - On "Specify Home Details" page, choose the same ORACLE_HOME as your previous installation.
 - Follow the instructions to complete the install.
2. Execute the Repository Wizard to upgrade the repository.
 - If you choose to run Repository Wizard during install, the Repository Wizard launches automatically.
 - Alternatively, you can launch Repository Wizard after installation. Refer to [Section 5.2.1.3, "Upgrade Your Mobile Server Repository"](#).
3. Execute the application upgrade assistant to upgrade user applications to the new repository. For more information, refer to [Section 5.2.1.4, "Upgrade Your Existing Applications"](#).

5.1.2.2 Upgrade in a separate ORACLE_HOME

Refer to [Table 5-1](#) and [Table 5-2](#) for the supported cases for upgrading Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 11g in a separate ORACLE_HOME.

1. Install Oracle Database Mobile Server 11g in a new ORACLE_HOME.
 - Launch Oracle Universal Installer. On "Select Installation Type" page, select "Mobile Server".
 - On "Specify Home Details" page, choose the same ORACLE_HOME that is different from your previous installation.
 - Follow the instructions to complete the install.

2. Execute the Repository Wizard to upgrade the repository.
 - If you choose to run Repository Wizard during install, the Repository Wizard launches automatically.
 - Alternatively, you can launch the Repository Wizard after installation. Refer to [Section 5.2.1.3, "Upgrade Your Mobile Server Repository"](#).
3. Execute the application upgrade assistant to upgrade user applications to the new repository. For more information, refer to [Section 5.2.1.4, "Upgrade Your Existing Applications"](#).
4. Uninstall the previous mobile server version from the old ORACLE_HOME.
 - Launch Oracle Universal Installer, click "Deinstall Products".
 - Choose the previous mobile server that you want to uninstall, and click "Remove".

5.1.2.3 Upgrade Your Mobile Server Repository

During installation of Oracle Database Mobile Server, the Repository Wizard detects if Oracle Database Lite 10.3.0.3.0 is installed and initiates the upgrade process, as follows:

Note: If you need to start the Repository Wizard independent of the install process, execute one of the following:

- Windows: %ORACLE_HOME%\Mobile\Server\admin\repwizard.bat
 - Unix: \$ORACLE_HOME/mobile/server/admin/repwizard
-
-

1. Enter the SYSTEM password. Click "Next".
2. Select the schema that you are going to upgrade. Click "Next".
3. Enter the password for the mobile server repository and click "YES" to install the sample applications. Click "Next".
4. A summary screen appears. Click "Next".
5. After the repository is upgraded, click "Finish".

In Oracle Database Mobile Server 11g, the repository location is set to %ORACLE_HOME%\Mobile\Server\admin\repository directory by default, which is different from Oracle Database Lite 10.3.0.3.0.

5.1.2.4 Upgrade Your Existing Applications

You can upgrade all existing applications with the Application Upgrade Assistant. Since the Oracle Database Mobile Server 11g repository is in a new location, the Application Upgrade Assistant moves applications from the old repository location to the new one.

The Application Upgrade Assistant is located as follows:

- On Windows: %ORACLE_HOME%\Mobile\Server\admin\ApplicationUpgradeAssistant.bat
- On Unix: \$ORACLE_HOME/mobile/server/admin/ApplicationUpgradeAssistant.sh

On the command-line, supply the following:

- Mobile server repository name and password
- The ORACLE_HOME for the old mobile server repository

For example, on a Windows system, where the repository schema name is **mobileadmin**, password is **manager**, the old ORACLE_HOME is **C:\Oracle\ora10gR3**, execute the Application Upgrade Assistant as follows:
ApplicationUpgradeAssistant.bat mobileadmin manager C:\Oracle\ora10gR3

The Application Upgrade Assistant will not upgrade applications for de-supported platforms, as follows:

- Applications deployed on de-supported platforms such as Oracle Lite, Web-to-Go, OC4J and BC4J platforms will not be upgraded.
- The default system folders and default application files for previous releases will not be upgraded, which includes admin-web, msadmin, olsetup, devmgr, META-INF, plugins, setup, and webtogo.
- In Oracle Database Mobile Server 11g, the Mobile Manager application is named 'mobile'. If you deployed an application with the same name in Oracle Database Lite 10.3.0.3.0, the Application Upgrade Assistant will not upgrade it due to a name conflict. You will need to manually change your application name and redeploy it on the new Mobile Server.

5.1.2.5 Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory (OID)

You can use the Oracle Internet Directory (OID), which is part of the Oracle Application Server, for storing and retrieving user information instead of the Mobile Server Repository. To use OID, you must migrate all user information from the existing repository into OID. For more information, refer to Section 4.3.1.6 "Managing OID Users in the Mobile Server" in the *Oracle Database Mobile Server Administration and Deployment Guide*.

5.1.3 Mobile Server Farm Upgrade

If you have multiple mobile servers in a farm that use the same mobile server repository on a back-end Oracle database, perform the following:

1. Shut down all mobile servers before you start the upgrade.
2. Upgrade one of these mobile servers and the mobile server repository.
3. Upgrade each additional mobile server that uses the repository.
4. Restart all mobile servers.

5.1.4 Upgrade Mobile Client

Berkeley DB and SQLite Mobile Clients installed with the Oracle Database Lite Mobile Server version 10.3.0.3.0 can be upgraded to Oracle Database Mobile Server 11g.

Table 5–4 shows what mobile client platforms can be upgraded to the Oracle Database Mobile Server 11g.

Table 5–4 Upgrading Mobile Clients

Client Device Platform	Upgrade Instructions
SQLite or Berkeley DB Mobile Clients on Win32	Run <code>update.exe</code> or <code>msync.exe</code> , which launches <code>update.exe</code> after the synchronization completes.

Table 5–4 (Cont.) Upgrading Mobile Clients

Client Device Platform	Upgrade Instructions
SQLite or Berkeley DB Mobile Clients on Pocket PC 6.0, 6.5	Run <code>update.exe</code> or <code>msync.exe</code> , which launches <code>update.exe</code> after the synchronization completes.
SQLite or Berkeley DB Mobile Clients on Pocket PC 5.0	Run <code>update.exe</code> or <code>msync.exe</code> , which launches <code>update.exe</code> after the synchronization completes. Only supported in upgrading to 11.2.0 or 11.3.0.
SQLite or Berkeley DB Mobile Clients on Linux	Run <code>update</code> or <code>msync</code> , which launches <code>update</code> after synchronization completes.
SQLite Mobile Client on Android	Run <code>update</code> .
SQLite Mobile Client on Blackberry	De-install the previous client version and install the appropriate 11g mobile client

To upgrade mobile client on Android, if server port is changed after server upgrade, modify `SERVER_URL` parameter in [NETWORK] section in `devmgr.ini` before running `update`. Usually you can find `devmgr.ini` at `/mnt/sdcard/oracle.mobileclient`.

To upgrade mobile clients on Win32, Windows Mobile and Linux platforms, follow the instructions below:

1. Shutdown your applications.
2. Check available software update and install it. Ensure that your device is enabled to receive software updates from the Mobile Server. You can either run `update.exe` or `msync.exe`, which automatically launches `update.exe` after the synchronization completes.

`Update.exe` might not work if the server port is changed after Mobile Server upgrade. In this case follow the instructions below to run `msync.exe`:

- Write `OSE.FILES=YES` in [DEFAULT] section in `ose.ini`. You can set `OSE.FILES=NO` if you want to disable `OSE.FILES` option after the client upgrade process completes.
 - Launch `msync.exe` and specify the new server URL in 'Server URL' field, click 'Sync' to perform synchronization. After synchronization completes, `update.exe` launches automatically and is able to connect to Mobile Server with the new server URL. `Update.exe` should find available software updates on the server and display the options for you to install the update.
3. After you update your mobile client, check if `NAME` and `TYPE` parameters are written in [DMC] section in `devmgr.ini`. If they are missing, copy them from the old configuration file `polite.ini`.
 4. If you are upgrading SQLite client, the client upgrade process is completed. If you are using the Berkeley DB client, double check if the old Berkeley DB environment files are removed during client upgrade. If not, follow the instructions to finish the upgrade process:
 - Shutdown your applications and `SyncAgent` if it is running.
 - Backup the Berkeley DB databases.
 - Remove environment files `__db.xxx` (`xxx` stands for 3 digits) in `*.db-journal` folders from where Berkeley DB databases are stored.
 - Restart your application and `SyncAgent` if needed.

5.1.4.1 Upgrade Oracle Mobile Clients from De-Supported Mobile Client Platforms

For Web-to-Go, OC4J, BC4J, and Branch Office Mobile Client platforms, there are no upgrade options since these client device platforms are no longer supported in Oracle Database Mobile Server 11g.

Table 5–5 shows how to migrate de-supported Mobile client platforms to available platforms in Oracle Database Mobile Server 11g Release 3.

Note: Do not cancel in the middle of the upgrade process.

Table 5–5 Upgrading from De-Supported Mobile Client Platforms

De-Supported Mobile Client Platform	Upgrade to the Available Platform
SQLite Mobile Client on Pocket PC 2003	
SQLite or Berkeley DB Mobile Clients on Pocket PC 5.0	SQLite or Berkeley DB Mobile Clients on Pocket PC 6.0

Note: SQLite and Berkeley DB Mobile Clients on Pocket PC 5.0 are supported in Oracle Database Mobile Server 11.2.0 and 11.3.0, but not supported in 11.1.0.

If your device is Windows Mobile 6.0 or later, you must perform the following after the CAB files are downloaded to the device:

1. Click on the CAB files to extract the files to the device.
2. Start the update by executing `update.exe`, which detects available upgrade options and prompts the user to upgrade.

If you do not upgrade mobile client devices for de-supported platforms, you can still synchronize data and perform administration duties with the following restrictions:

- You can continue to administer de-supported client platforms, and they will appear in the platform tabs in the Mobile Manager. However, these platforms will not be available on the setup page for you to create new clients using these platforms.
- The applications that already exist on the de-supported client platforms will still execute as expected. You may also apply any patches for the application. However, you cannot deploy any new applications to that platform.
- If you do upgrade the client platform, you must re-package and re-publish each application as a new application for the new client platform.

5.2 Upgrade Oracle Database Mobile Server 11.1.0 or 11.2.0 to 11.3.0

The following sections describe the steps for upgrading your software:

- [Section 5.2.1, "Upgrade Mobile Server"](#)
- [Section 5.2.2, "Upgrade Mobile Client"](#)
- [Section 5.2.3, "Mobile Server Farm Upgrade"](#)

5.2.1 Upgrade Mobile Server

Upgrading Oracle Database Mobile Server 11.1.0 or 11.2.0 to Oracle Database Mobile Server 11.3.0 can be performed on an installation in the same ORACLE_HOME or in a separate ORACLE_HOME. You must apply all patches before upgrading from Oracle Database Mobile Server 11.1.0 or 11.2.0 to 11.3.0. Table 5-6 shows what version of Oracle Database Mobile Server 11.1.0/11.2.0 and application server installation can be upgraded to Oracle Database Mobile Server 11.3.0.

The following sections describe how to perform the upgrade:

- [Section 5.2.1.1, "Upgrading in the Same ORACLE_HOME"](#)
- [Section 5.2.1.2, "Upgrading in a Separate ORACLE_HOME"](#)
- [Section 5.2.1.3, "Upgrade Your Mobile Server Repository"](#)
- [Section 5.2.1.4, "Upgrade Your Existing Applications"](#)
- [Section 5.2.1.5, "Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory \(OID\)"](#)

Table 5–6 Upgrading Oracle Database Mobile Server 11.1.0 or 11.2.0 to 11.3.0

From Oracle Database Mobile Server 11.1.0 or 11.2.0	To Oracle Database Mobile Server 11.3.0 Base	Same Oracle Home	Separate Oracle Home
Mobile Server 11.1.0 or 11.2.0 with Oracle WebLogic Server 11g Release 1, Mobile Server 11.2.0 with Oracle WebLogic Server 12c	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c	Supported	Supported
	GlassFish 3.1	Supported	Supported
	Oracle iAS 10.1.3.5.0	Supported	Supported
	Apache TomEE 1.0	Supported	Supported
Mobile Server 11.1.0 or 11.2.0 with GlassFish 3.1	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c	Supported	Supported
	GlassFish 3.1	Supported	Supported
	Oracle iAS 10.1.3.5.0	Supported	Supported
	Apache TomEE 1.0	Supported	Supported
Mobile Server 11.2.0 with Apache TomEE 1.0	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c	Supported	Supported
	GlassFish 3.1	Supported	Supported
	Oracle iAS 10.1.3.5.0	Supported	Supported
	Apache TomEE 1.0	Supported	Supported

5.2.1.1 Upgrading in the Same ORACLE_HOME

Refer to [Section 5.1.2.1, "Upgrade in the same ORACLE_HOME"](#).

5.2.1.2 Upgrading in a Separate ORACLE_HOME

Refer to [Section 5.1.2.2, "Upgrade in a separate ORACLE_HOME"](#).

5.2.1.3 Upgrade Your Mobile Server Repository

Refer to [Section 5.1.2.3, "Upgrade Your Mobile Server Repository"](#).

5.2.1.4 Upgrade Your Existing Applications

You can upgrade all existing applications with the Application Upgrade Assistant.

The Application Upgrade Assistant is located as follows:

- On Windows: %ORACLE_
HOME%\Mobile\Server\admin\ApplicationUpgradeAssistant.bat
- On Unix: \$ORACLE_
HOME/mobile/server/admin/ApplicationUpgradeAssistant.sh

On the command-line, supply the following:

- Mobile server repository name and password
- The ORACLE_HOME for the old mobile server repository

For example, on a Windows system, where the repository schema name is **mobileadmin**, password is **manager**, the old ORACLE_HOME is **C:\Oracle\MobileServer11g**, execute the Application Upgrade Assistant as follows:
ApplicationUpgradeAssistant.bat mobileadmin manager C:\Oracle\MobileServer11g

5.2.1.5 Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory (OID)

Refer to [Section 5.1.2.5, "Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory \(OID\)"](#).

5.2.2 Upgrade Mobile Client

Berkeley DB and SQLite Mobile Clients installed with the Oracle Database Mobile Server 11.1.0 or 11.2.0 can be upgraded to version 11.3.0. [Table 5–7](#) shows what mobile client platforms can be upgraded to the Oracle Database Mobile Server 11.3.

Table 5–7 Upgrading Mobile Clients

Client Device Platform	Upgrade Instructions
SQLite or Berkeley DB Mobile Clients on Win32	Run <code>update.exe</code> or <code>msync.exe</code> , which launches <code>update.exe</code> after the synchronization completes.
SQLite or Berkeley DB Mobile Clients on Pocket PC 5.0, 6.0, 6.5	Run <code>update.exe</code> or <code>msync.exe</code> , which launches <code>update.exe</code> after the synchronization completes.
SQLite or Berkeley DB Mobile Clients on Linux	Run <code>update</code> or <code>msync</code> , which launches <code>update</code> after synchronization completes.
SQLite Mobile Client on Android	Run <code>update</code> .
SQLite Mobile Client on Blackberry	De-install the previous client version and install the latest version.

Refer to [Section 5.1.4, "Upgrade Mobile Client"](#) for the detailed upgrade instructions.

The following section provides information on upgrading Berkeley DB mobile client to 11.3.0:

- [Section 5.2.2.1, "Upgrade Berkeley DB Mobile Client to 11.3.0"](#)

5.2.2.1 Upgrade Berkeley DB Mobile Client to 11.3.0

In Oracle Database Mobile Server 11.3.0, Berkeley DB SQL libraries are upgraded to version 6.0, and bundled with Berkeley DB Mobile Client 11.3.0. Since the binary format of Berkeley DB 6.0 is not compatible with that of previous Berkeley DB releases, a direct upgrade of Berkeley DB databases would not work. Hence, there are two options to upgrade Berkeley DB databases:

- Backup and dump Berkeley DB databases before upgrade and re-create Berkeley DB 6.0 databases after upgrade.

For this option, follow the steps below:

1. Dump old Berkeley DB databases (*.db) stored under data directory before Berkeley DB Mobile Client upgrade. For example, if you have 4 databases under data directory, use .dump command of old dbsql to dump the databases into SQL text files as follows:

```
prompt>dbsql.exe ..\bdb\data\JOHN\test.db .dump > test.sql
prompt>dbsql.exe ..\bdb\data\JOHN\default.db .dump > default.sql
prompt>dbsql.exe ..\bdb\data\JOHN\OSE$test.db .dump > OSE$test.sql
prompt>dbsql.exe ..\bdb\data\JOHN\OSE$default.db .dump > OSE$default.sql
```

You may run out of space of the mobile device where BDB Mobile Client is installed. In this case, you can copy old Berkeley DB databases, dbsql tool and Berkeley DB libraries to a separate system (maybe call it dump system for example) and do Step 1 in dump system.

2. Remove all the files under data directory, including *.db and *.db-journal. In this example, remove all files under bdb\data\JOHN.
3. Upgrade Mobile Server and Berkeley DB Mobile Client to 11.3.0.
4. Re-create Berkeley DB databases from the SQL text files with dbsql 6.0.

```
prompt>dbsql.exe ..\bdb\data\JOHN\test.db < test.sql
prompt>dbsql.exe ..\bdb\data\JOHN\default.db < default.sql
prompt>dbsql.exe ..\bdb\data\JOHN\OSE$test.db < OSE$test.sql
prompt>dbsql.exe ..\bdb\data\JOHN\OSE$default.db < OSE$default.sql
```

If you do Step 1 and Step 4 in a dump system, you need to copy the new Berkeley DB databases back to the data directory of Berkeley DB Mobile Client in the mobile device.

- Use Mobile Server repository for data backup.

For this option, follow the steps below:

1. Before the upgrade, synchronize Berkeley DB Mobile Client with Mobile Server to ensure that all client data is saved in the Mobile Server repository.
2. Uninstall old Berkeley DB Mobile Client. You can either uninstall the Mobile Client manually, or send the 'uninstall' command from Mobile Server to uninstall Mobile Client on Win32, Windows Mobile and Linux platforms automatically.
3. Upgrade Mobile Server to 11.3.0 and install Berkeley DB Mobile Client 11.3.0.
4. Do a sync with the new Berkeley DB Mobile Client. Berkeley DB 6.0 databases will be created under data directory.

5.2.3 Mobile Server Farm Upgrade

Refer to [Section 5.1.3, "Mobile Server Farm Upgrade"](#).

5.3 Upgrade Oracle Database Mobile Server 11.1.0 to 11.2.0

The upgrade from Oracle Database Mobile Server 11g Release 1 to 11g Release 2 is straightforward. You can upgrade Oracle Database Mobile Server either in the same ORACLE_HOME or in a separate ORACLE_HOME. [Table 5–8](#) shows what software is supported when upgrade from an Oracle Database Mobile Server 11g Release 1 and its installed application server to Oracle Database Mobile Server 11g Release 2 and supported application servers.

Table 5–8 Upgrading Oracle Database Mobile Server 11.1.0.0.0 to 11.2.0.0.0

From Oracle Database Mobile Server 11g Release 1	To Oracle Database Mobile Server 11g Release 2	Same Oracle Home	Separate Oracle Home
Oracle Database Mobile Server 11.1.0.0.0 with GlassFish 3.1	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c	Supported	Supported
	GlassFish 3.1	Supported	Supported
	Oracle iAS 10.1.3.5.0	Supported	Supported
	Apache TomEE 1.0	Supported	Supported
Oracle Database Mobile Server 11.1.0.0.0 with WebLogic 11g Release 1	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c	Supported	Supported
	GlassFish 3.1	Supported	Supported
	Oracle iAS 10.1.3.5.0	Supported	Supported
	Apache TomEE 1.0	Supported	Supported
Oracle Database Mobile Server 11.1 or 11.2 with Oracle iAS 10.1.3.5.0	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c	Supported	Supported
	GlassFish 3.1	Supported	Supported
	Oracle iAS 10.1.3.5.0	Supported	Supported
	Apache TomEE 1.5	Supported	Supported
Oracle Database Mobile Server 11.1.0.0.0 with Oracle iAS 10.1.3.5.0	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c	Supported	Supported
	GlassFish 3.1	Supported	Supported
	Oracle iAS 10.1.3.5.0	Supported	Supported
	Apache TomEE 1.0	Supported	Supported

5.3.1 Mobile Server Farm Upgrade

Refer to [Section 5.1.3, "Mobile Server Farm Upgrade"](#) for instructions on how to upgrade Mobile Server in a farm.

5.3.2 Upgrade Oracle Database Mobile Server 11.1.0.0.0 to 11.2.0.0.0

The following sections describe how to upgrade from Oracle Database Mobile Server 11.1 or 11.2 to Oracle Database Mobile Server 11.2.0.0.0:

- [Section 5.3.2.1, "Upgrade in the Same or a Separate ORACLE_HOME"](#)
- [Section 5.3.2.2, "Upgrade Your Mobile Server Repository"](#)

5.3.2.1 Upgrade in the Same or a Separate ORACLE_HOME

Upgrading Oracle Database Mobile Server 11.1.0.0.0 to 11.2.0.0.0 is supported in the same or in a separate ORACLE_HOME. For more information, see:

- [Section 5.3.2.1.1, "Upgrading in the Same ORACLE_HOME"](#)
- [Section 5.3.2.1.2, "Upgrading in a Separate ORACLE_HOME"](#)

5.3.2.1.1 Upgrading in the Same ORACLE_HOME

Refer to [Table 5–8](#) for the supported cases for upgrading Oracle Database Mobile Server 11.1.0.0.0 to 11.2.0.0.0 in the same ORACLE_HOME.

1. Install Oracle Database Mobile Server 11.2.0.0.0 in the same ORACLE_HOME.
 - Launch Oracle Universal Installer. On Select Installation page, select "Mobile Server".
 - On the "Specify Home Details" page, choose the same ORACLE_HOME as your previous installation.
 - Follow the instructions to complete the install.
2. Execute the Repository Wizard to upgrade the repository.
 - If you choose to run Repository Wizard during install, the Repository Wizard launches automatically.
 - Alternatively, you can launch the Repository Wizard after installation. Refer to [Section 5.1.2.3, "Upgrade Your Mobile Server Repository"](#).

5.3.2.1.2 Upgrading in a Separate ORACLE_HOME

Refer to [Table 5–8](#) for the supported cases for upgrading Oracle Database Mobile Server 11.1.0.0.0 to 11.2.0.0.0 in a separate ORACLE_HOME.

1. Install Oracle Database Mobile Server 11.2.0.0.0 in a new ORACLE_HOME.
 - Launch Oracle Universal Installer, on Select Installation Type page, select "Mobile Server".
 - On the "Specify Home Details" page, choose a new ORACLE_HOME that is different from your previous installation.
 - Follow the instructions to complete the install.
2. Execute the Repository Wizard to upgrade the repository.
 - If you choose to run Repository Wizard during install, the Repository Wizard launches automatically.
 - Alternatively, you can launch Repository Wizard after installation. Refer to [Section 5.1.2.3, "Upgrade Your Mobile Server Repository"](#).
3. Execute the application upgrade assistant to upgrade user applications to the new repository. For more information, refer to [Section 5.3.3, "Upgrade Your Existing Applications"](#).
4. Uninstall the previous Mobile Server from the old ORACLE_HOME.
 - Launch Oracle Universal Installer, click "Deinstall Products".
 - Choose the previous mobile server that you want to uninstall, and click "Remove".

5.3.2 Upgrade Your Mobile Server Repository

For information on upgrading the mobile server repository, refer to [Section 5.1.2.3, "Upgrade Your Mobile Server Repository"](#).

5.3.3 Upgrade Your Existing Applications

For information on upgrading existing applications, refer to [Section 5.3.3, "Upgrade Your Existing Applications"](#). update.exe or msync.exe

5.3.4 Upgrade Mobile Clients

Berkeley DB and SQLite Mobile Clients installed with the Oracle Database Mobile Server 11.1.0.0.0 can be upgraded to 11.2.0.0.0. [Table 5–9](#) shows the mobile client platforms that can be upgraded to 11.2.0.0.0.

Table 5–9 Upgrading Mobile Clients

Client Device Platform	Upgrade Instructions
SQLite or Berkeley DB Mobile Clients on Win32	Run update.exe or msync.exe, which launches update.exe after the synchronization completes.
SQLite or Berkeley DB Mobile Clients on Pocket PC 6.0, 6.5	Run update.exe or msync.exe, which launches update.exe after the synchronization completes.
SQLite or Berkeley DB Mobile Clients on Linux	Run update or msync, which launches update after the synchronization completes.
SQLite Mobile Clients on Android	Run update.
SQLite Mobile Clients on BlackBerry	De-install the previous client version and install the 11.2 mobile client.

To upgrade mobile client on Android, if server port is changed after server upgrade, please modify SERVER_URL parameter in [NETWORK] section in devmgr.ini before running update. Usually you can find **devmgr.ini at /mnt/sdcard/oracle.mobileclient**

To upgrade mobile clients on Win32, Windows Mobile and Linux platforms, follow the instructions below:

1. Shutdown your applications.
2. Check available software update. Ensure that your device is enabled to receive software updates from the Mobile Server. You can either run update.exe or msync.exe, which automatically launches update.exe after the synchronization completes.

Update.exe might not work if the server port is changed after Mobile Server upgrade. In this case follow the instructions below to run msync.exe:

- Run msync.exe and specify the new server URL in "Server URL" field, so that the update.exe is able to connect to Mobile Server with the new server URL.
The update.exe will find the available software updates on the server and display the option for you to install the update.
3. Install the software update. A new mobile client would be downloaded and installed.
 4. Restart you applications.

5.4 Upgrade Oracle Database Lite from 10g Release 1 or 2 to 10g Release 3

If you have Oracle Database Lite 10g Release 1 or 2, you must upgrade to the latest version. The following sections describe the steps for upgrading your software:

- [Section 5.4.1, "Supported Versions for Upgrading Oracle Database Lite"](#)
- [Section 5.4.2, "Pre-Upgrade Instructions"](#)
- [Section 5.4.3, "Upgrade Your Mobile Server Repository"](#)
- [Section 5.4.4, "Upgrade Your Existing Applications"](#)
- [Section 5.4.5, "Installing Sample Applications"](#)
- [Section 5.4.6, "Migrate Your Users From the Mobile Server Repository to the Oracle Internet Directory \(OID\)"](#)
- [Section 5.4.7, "Remove the 10g Release 1 or 2 Installation"](#)
- [Section 5.4.8, "Remove Duplicate Mobile Server IDs from the Repository"](#)

5.4.1 Supported Versions for Upgrading Oracle Database Lite

Upgrading Oracle Database Lite from 10g Release 1 or 2 to 10g Release 3 is supported in the same or a separate `ORACLE_HOME` from the previous installation. Before you upgrade your Mobile Server from Oracle Database Lite 10g Release 2 to Release 3, be sure to apply all patches first.

The following sections describe how to perform the upgrade:

- [Section 5.4.1.1, "Upgrading in the Same ORACLE_HOME"](#)
- [Section 5.4.1.2, "Upgrading in a Separate ORACLE_HOME"](#)
- [Section 5.4.1.3, "Supported Oracle Database Lite Versions for Upgrade"](#)

5.4.1.1 Upgrading in the Same ORACLE_HOME

The Oracle Universal Installer detects the earlier version and uninstalls this version. All deployed application files and existing configuration files remain to be used by the new installation. When prompted, you can re-execute the Repository Wizard, which detects that the repository exists and runs the repository upgrade instead of the install.

5.4.1.2 Upgrading in a Separate ORACLE_HOME

These steps are the standard steps to follow when upgrading a Mobile Server in a separate `ORACLE_HOME` directory.

1. Install the desired version of Oracle Application Server into the new `ORACLE_HOME`. This step is optional.
2. Install Mobile Server in the new `ORACLE_HOME`.
3. Execute the Repository Wizard to upgrade the repository.
4. Execute the application upgrade assistant.
5. Uninstall the previous Mobile Server version from the old `ORACLE_HOME`.

5.4.1.3 Supported Oracle Database Lite Versions for Upgrade

[Table 5–10](#) details what versions of Oracle Database Lite and OracleAS installation can be upgraded to Oracle Database Lite 10.3.0.3

Table 5–10 Upgrading Oracle Database Lite 10.1.x to Oracle Database Lite 10.3.0.3

From <Version>	To Oracle Database Lite 10.3.0.3 Base	Same Oracle Home	Separate Oracle Home
10g R1 Standalone	Oracle iAS 10.1.3.5.0	Not Supported	Supported
	Oracle WebLogic Server 11g Rel 1, Oracle WebLogic Server 12c	Not Supported	Supported
10g R1 with Oracle iAS 9.0.2	Any	Not Supported	Not Supported
10g R1 with Oracle iAS 9.0.3	Any	Not Supported	Not Supported
10g R1 with Oracle iAS 9.0.4	Any	Not Supported	Not Supported
10g R1 with Oracle iAS 10.1.2 with patch 5723922	Oracle iAS 10.1.3.5.0	Not Supported	Supported
	Oracle WebLogic Server 11g Rel 1, Oracle WebLogic Server 12c	Not Supported	Supported

For Oracle Database Lite 10.0, support was removed for installations with Oracle iAS 9.0.2, 9.0.3 and 9.0.4. In addition, the OC4J version in the standalone installation is Oracle iAS version 9.0.4. You can apply the patch that enables support for Oracle iAS 10.1.2 and upgrades the version of standalone OC4J to Oracle iAS 10.1.2. The patch is 5723922: Needs migration path from Oracle iAS and OC4J 9.0.4.x to Oracle iAS 10.1.2.

In Oracle Database Lite 10.0, the path used was `mobile/j2ee/home`. However, in Oracle Database Lite 10.3, this was modified to use the `mobile/j2ee/mobileserver` path. This may cause problems with certain Oracle iAS upgrades.

Table 5–11 Upgrading Oracle Database Lite 10.2.x to Oracle Database Lite 10.3.0.3

From <Version>	To Oracle Database Lite 10.3.0.3 Base	Same Oracle Home	Separate Oracle Home
10g R2 Standalone	Oracle iAS 10.1.3.5.0	Not Supported	Supported
	Oracle WebLogic Server 11g Rel 1, Oracle WebLogic Server 12c	Not Supported	Supported
10g R2 with Oracle iAS 9.0.4	Oracle iAS 10.1.3.5.0	Not Supported	Not Supported
	Oracle WebLogic Server 11g Rel 1, Oracle WebLogic Server 12c	Not Supported	Not Supported
10g R2 with Oracle iAS 10.1.2 with patch 5723922	Oracle iAS 10.1.3.5.0	Not Supported	Supported
	Oracle WebLogic Server 11g Rel 1, Oracle WebLogic Server 12c	Not Supported	Supported

For Oracle Database Lite 10.2, support was removed for installations with Oracle iAS 9.0.4. In addition, the OC4J version in the standalone installation is Oracle iAS version 9.0.4. You can apply the patch that enables support for Oracle iAS 10.1.2 and upgrades

the version of standalone OC4J to Oracle *iAS* 10.1.2. The patch is 5723922: Needs migration path from Oracle *iAS* and OC4J 9.0.4.x to Oracle *iAS* 10.1.2.

Table 5–12 Upgrading Oracle Database Lite 10g Release 3 to Oracle Database Lite 10.3.0.3

From <Version>	To Oracle Database Lite 10.3.0.3 Base	Same Oracle Home	Separate Oracle Home
10.3.0.1 Standalone	Oracle <i>iAS</i> 10.1.3.5.0	Not Supported	Supported
	Oracle WebLogic Server 11g Rel 1, Oracle WebLogic Server 12c	Not Supported	Supported
10.3.0.1 with Oracle <i>iAS</i> 10.1.2	Standalone	Not Supported	Supported
	Oracle <i>iAS</i> 10.1.2.0.2	Supported	Supported
	Oracle <i>iAS</i> 10.1.3.1.0	Not Supported	Supported
10.3.0.1 with Oracle <i>iAS</i> 10.1.3	Oracle <i>iAS</i> 10.1.3.5.0	Supported	Supported
	Oracle WebLogic Server 11g Rel 1, Oracle WebLogic Server 12c	Supported	Supported
10.3.0.1 with Oracle <i>iAS</i> 10.1.3.1	Oracle <i>iAS</i> 10.1.3.5.0	Supported	Supported
	Oracle WebLogic Server 11g Rel 1, Oracle WebLogic Server 12c	Supported	Supported

Table 5–13 Upgrading Oracle Database Lite with OID to Oracle Database Lite 10.3.0.3

From <Version>	From OID <Version>	To OID <Version>	Supported?
5.0.2.10	Oracle <i>iAS</i> 9.0.2	Any	Not Supported
10.0.0	Oracle <i>iAS</i> 9.0.2	Any	Not Supported
	Oracle <i>iAS</i> 9.0.3	Any	Not Supported
	Oracle <i>iAS</i> 9.0.4	Any	Not Supported
	Oracle <i>iAS</i> 10.1.2.	Oracle <i>iAS</i> 10.1.2 with OID 10.1.2	Supported
10.2.0.2	Oracle <i>iAS</i> 9.0.4	Any	Not Supported
	Oracle <i>iAS</i> 10.1.2 with OID 10.1.2	Oracle <i>iAS</i> 10.1.2.0.2 with OID 10.1.2	Supported
10.3.0.1	Oracle <i>iAS</i> 10.1.2 with OID 10.1.2	Oracle <i>iAS</i> 10.1.2 with OID 10.1.2	Supported
	Oracle <i>iAS</i> 10.1.3.1 with OID 10.1.4	Oracle <i>iAS</i> 10.1.3.x with OID 10.1.4	Supported

5.4.2 Pre-Upgrade Instructions

In order to upgrade your Oracle Database Lite Installation, perform the following:

1. If you have multiple mobile servers that use the same mobile server repository on a back-end database, you must shut down all of these mobile servers before you start the upgrade. Then, upgrade one of these mobile servers and the mobile server repository. After this upgrade completes successfully, you can upgrade each additional mobile server that uses the repository. After all of the shared mobile servers are upgraded, you can restart them.
2. You can upgrade your existing Database Lite 10g Release 1 or 2 to the Release 3 in the same Oracle Home OR you can install the Release 2 of Oracle Database Lite in a separate Oracle home than your Database Lite 10g Release 1 or 2 home.

See [Chapter 4, "Installation of Oracle Database Mobile Server"](#) for instructions on how to install Oracle Database Lite. When the Repository Wizard is executed, see [Chapter 5.4.3, "Upgrade Your Mobile Server Repository"](#) for directions on how to upgrade the repository on the back-end database.

3. When you migrate any of your mobile servers to an instance running on a WebLogic Server, the mobile server URL and port may change. Each client application, including `msync.exe`, that interacts with these mobile servers must apply the new server URL on the client side.

When the mobile server URL change occurs, perform the following on each client:

- a. Stop all processes on the client.
- b. Modify the URL on the client by opening `msync.exe`, change the server URL, and apply the changes.
- c. Perform a synchronization.

5.4.3 Upgrade Your Mobile Server Repository

During installation of Oracle Database Lite 10g Release 3, the Repository Wizard detects if Release 1 or 2 is installed and initiates the upgrade process, as follows:

Note: If you need to start the Repository Wizard outside of the install process, execute

```
ORACLE_HOME\Mobile\Server\admin\repwizard
```

1. Enter the `SYSTEM` password. Click "Next".
2. Select the schema that you are going to upgrade. Click "Next".
3. Enter the password for the Mobile Server Repository—`MOBILEADMIN`—and click "No" for installing the sample applications. Click "Next".
4. A summary screen appears. Click "Next".
5. After the repository is upgraded, click "Finish".

To upgrade each individual mobile server, follow the instructions in [Section 5.1.1, "Pre-Upgrade Instructions"](#).

5.4.4 Upgrade Your Existing Applications

You have to upgrade existing applications only if you installed Oracle Database Lite 10g Release 3 in a separate `ORACLE_HOME` than Oracle Database Lite 10g Release 1 or 2.

Thus, once you have installed the Oracle Database Lite 10g Release 3 in a new *ORACLE_HOME* and the Repository Wizard has upgraded the mobile server repository successfully, execute the Application Upgrade Assistant to upgrade your existing applications.

The Application Upgrade Assistant moves applications from the *ORACLE_HOME* for Oracle Database Lite 10g Release 1 or 2 to your the new *ORACLE_HOME* for Oracle Database Lite 10g Release 3.

The Application Upgrade Assistant is located in the following locations:

- On a Windows environment:

```
ORACLE_HOME\mobile\server\admin\ApplicationUpgradeAssistant.bat
```

- On a UNIX environment:

```
ORACLE_HOME/mobile/server/admin/ApplicationUpgradeAssistant.sh
```

On the command-line, supply the following:

- Mobile server repository name and password
- The old *ORACLE_HOME*
- The new *ORACLE_HOME*

For example, on a Windows system, where the repository name is *mobileadmin*, password is *manager*, the old *ORACLE_HOME* is *c:\oracle\ora10gR1* and the new *ORACLE_HOME* is *c:\oracle\ora10gR2*, execute the Application Upgrade Assistant, as follows:

```
ApplicationUpgradeAssistant mobileadmin manager
c:\oracle\ora10gR1 c:\oracle\ora10gR2
```

5.4.5 Installing Sample Applications

For the sample applications, you can either upgrade the existing sample applications, use the existing samples in Release 3, or install new sample applications. The Release 1 or 2 samples will still work in Release 3 without any upgrade.

To install samples use following command:

```
demoinstaller -install
[Database_Administrator_User_Name] [Database_Administrator_Password]
[Repository_Owner] [Repository_Password]
[Demo_user_name ][Demo_password]
```

For Example:

```
demoinstaller -install system manager mobileadmin manager master master
```

The *demoinstaller* utility is located in the following directories:

- On a Windows environment:

```
ORACLE_HOME\mobile\server\demos\demoinstaller.bat
```

- On a UNIX environment:

```
ORACLE_HOME/mobile/server/demos/demoinstaller.sh
```

5.4.6 Migrate Your Users From the Mobile Server Repository to the Oracle Internet Directory (OID)

You can use the Oracle Internet Directory (OID), which is part of the Oracle application server, for storing and retrieving user information instead of the mobile server repository. To use OID, you must migrate all user information from the existing repository into OID.

When you migrate users from a mobile server repository into OID, you cannot have duplicate users in OID. So, if you migrate users from two repositories into a single OID, and you have users with the same name, but different passwords on two separate repositories, the user that is first migrated into OID is the one that is valid. The second attempt to migrate an existing username into OID from a different repository will not migrate and no message is provided. This can be a problem if you have two users in different repositories with different passwords.

Migrate existing users in the repository to OID through the `oiduser` tool, which is located in `ORACLE_HOME\Mobile\Server\bin`. The `oiduser` tool migrates existing users with either randomly-generated passwords or a common password.

The following sections describe how to migrate your users based on which Oracle Application Server you have installed:

- [Section 5.4.6.1, "Mobile Server Installed On Oracle Application Server 10.1.2.0.0"](#)
- [Section 5.4.6.2, "Mobile Server Installed on Oracle Application Server 10.1.3.1.0 or Higher"](#)

5.4.6.1 Mobile Server Installed On Oracle Application Server 10.1.2.0.0

Perform the following to migrate your users to OID:

1. Set the `IAS_MODE` parameter in the `webtogo.ora` file to `YES`.
2. If you are using Oracle9iAS, then explicitly grant permission to the `webtogo.jar` file, which enable calls originating from this JAR file. Grant this permission by adding the following to the `jazn-data.xml` file, which is located in the `ORACLE_HOME/config` directory:

```
<grant>
  <grantee>
    <codesource>
      <url>file:$MIDTIER_ORACLE_HOME/mobile/server/bin/webtogo.jar</url>
    </codesource>
  </grantee>
  <permissions>
    <permission>
      <class>oracle.ias.repository.schemaimpl.CheckRepositoryPermission</class>
      <name>connectAs</name>
    </permission>
  </permissions>
</grant>
```

3. Migrate the user information using the `oiduser` tool, for either randomly-generated passwords or a common password, as follows:
 - To use randomly-generated passwords for each user, execute the `oiduser` tool without the `-P` option, as follows:

```
oiduser <ORACLE_HOME> <Mobile Server Repository username> <Mobile
Server Repository password> <OID port number> <OID host name> <OID
password> <OID admin name> <OID subscriber name>
```

For example, the default setting would be:

```
oiduser <ORACLE_HOME> mobileadmin manager 389 ldap://myhost-pc1.com
welcome1 orcladmin dc=us,dc=oracle,dc=com
```

- To use a common password for all users, provide the common password with the `-P` option, as follows:

```
oiduser <ORACLE_HOME> <Mobile Server Repository username> <Mobile
Server Repository password> <-P> <common password> <OID port number> <OID
host name> <OID password> <OID admin name> <OID subscriber name>
```

where the common password is specified by you.

4. The `oiduser` tool generates the `LDAP.bat`, `oiduser.dat` and `mobile_oid_user` files. Copy these files to the same directory in the application infrastructure machine where OID is installed. You can copy them to any directory, as the `LDAP.bat` executable uses `oiduser.dat` and the `mobile_oid_user` executable uses the generated `oiduserfile.Idif` file.

Note: The `mobile_oid_user` executable requires that the `ldapadd` executable is in the `PATH`. The `ldapadd` executable is part of the application server installation.

5. Execute the `LDAP.bat` from the command-line with the `oiduser.dat` as input on the machine where the application server resides. This creates the `oiduserfile.Idif` file.
6. Execute the `mobile_oid_user` file from the command-line on the application server machine. This creates the Oracle Database Lite users in the OID.

All users from the Mobile Server Repository are now migrated to the OID with the passwords, as designated in step 1.

Perform the following to enable user authentication from OID on the Mobile Server:

1. Login to Mobile Manager as the administrator and select the appropriate server.
2. Click on the Administration tab.
3. Click "Edit Config File" to edit the `webtogo.ora` file for this server.
4. If `SSO_ENABLED` has a hash mark (#) before it, then eliminate the hash mark and set `SSO_ENABLED` to `YES`. Click "Apply".
5. Restart both the application server and the Mobile Server.

5.4.6.2 Mobile Server Installed on Oracle Application Server 10.1.3.1.0 or Higher

Perform the following to migrate your users to OID:

1. Set the `IAS_MODE` parameter in the `webtogo.ora` file to `YES`.
2. Migrate the user information using the `oiduser` tool, for either randomly-generated passwords or a common password, as follows:
 - To use randomly-generated passwords for each user, execute the `oiduser` tool without the `-P` option, as follows:

```
oiduser <ORACLE_HOME> <Mobile Server Repository username> <Mobile
Server Repository password> <OID port number> <OID host name> <OID
password> <OID admin name> <OID subscriber name>
```

For example, the default setting would be:

```
oiduser <ORACLE_HOME> mobileadmin manager 389 ldap://myhost-pc1.com
welcome1 orcladmin dc=us,dc=oracle,dc=com
```

- To use a common password for all users, provide the common password with the `-P` option, as follows:

```
oiduser <ORACLE_HOME> <Mobile Server Repository username> <Mobile
Server Repository password> <-P> <common password> <OID port number> <OID
host name> <OID password> <OID admin name> <OID subscriber name>
```

where the common password is specified by you.

All users from the Mobile Server Repository are now migrated to the OID with the required passwords.

If you want to enable Oracle Single Sign on on the Mobile Server then perform the following:

1. Login to Mobile Manager as the administrator and select the appropriate server.
2. Click on the Administration tab.
3. Click "Edit Config File" to edit the `webtogo.ora` file for this server.
4. If `SSO_ENABLED` has a hash mark (#) before it, then eliminate the hash mark and set `SSO_ENABLED` to YES. Click "Apply".
5. Restart both the application server and the Mobile Server.

5.4.7 Remove the 10g Release 1 or 2 Installation

Once you have completed all of the upgrade exercises, you can now remove the Oracle Database Lite 10g Release 1 or 2 from your system. Only perform these steps if you have installed Oracle Database Lite 10g Release 3 in a separate `ORACLE_HOME`. In addition, this is not required, but is recommended for clarity.

1. Start the Oracle Universal Installer by double-clicking on `setup.exe`.
2. On the File Locations screen, enter the `ORACLE_HOME` and `PATH` for the 10g Release 1 or 2 installation. Click "Installed Products".
3. On the Inventory screen, select the '+' next to the `ORACLE_HOME` for your 10g Release 1 or 2 installation. Then, click the box next to the installation that you are going to remove. Click "Reove".
4. On the Confirmation screen, click "Yes".
5. When the removal is complete, click "Close".
6. To exit the Installer, click "Exit".

5.4.8 Remove Duplicate Mobile Server IDs from the Repository

After migrating Oracle Database Lite 10g Release 1 or 2 to Release 3, you will see duplicate entries in the Mobile Manager Farms page. To delete these entries from the repository, use the `unregistermobileserver` utility:

- The syntax for `unregistermobileserver` utility prior to Oracle Database Mobile Server 11.3.0.1 release is:

```
unregistermobileserver [Repository_Name] [Repository_Password] [MobileServer_
id]
```

- The syntax for `unregistermobileserver` utility since Oracle Database Mobile Server 11.3.0.1 release is:

```
unregistermobileserver [Repository_Name]/[Repository_Password] [MobileServer_id]
```

To get the old `MobileServer_id` that needs to be removed:

- Connect to the Mobile Server repository with `Repository_Name` and `Repository_Password`.
- Execute the SQL script to find out the old Mobile Server id:

```
Select id from mobileserver_instances where version <> 'new_version';
```

The `new_version` is the Mobile Server version you are running after the upgrade, for example, 11.3.0.0.0.

Following is the location of `unregistermobileserver`:

- On a Windows environment:

```
ORACLE_HOME\mobile\server\admin\unregistermobileserver.bat
```
- On a UNIX environment:

```
ORACLE_HOME/mobile/server/admin/unregistermobileserver.sh
```

5.5 Upgrade Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 12c

Oracle Database Mobile Server 12c contains most of the components of the previous Oracle Database Lite product, with a few exceptions. The Oracle Lite client is no longer supported. Instead, you must choose either a Berkeley DB or SQLite client. Thus, if you have been using Oracle Database Lite in the past, all Oracle Database Lite Mobile Clients will need to be migrated to either of these supported mobile clients. In addition, your Mobile Server needs to be upgraded to the mobile server 12c version. You can upgrade your Mobile Server from 10.3.0.3.0 to 12.1.0 directly.

The following sections detail how you can upgrade Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 12c for both your existing mobile server and mobile client.

- [Section 5.5.1, "Pre-Upgrade Instructions"](#)
- [Section 5.5.2, "Upgrade Mobile Server"](#)
- [Section 5.5.3, "Mobile Server Farm Upgrade"](#)
- [Section 5.5.4, "Upgrade Mobile Client"](#)

5.5.1 Pre-Upgrade Instructions

You can only upgrade an Oracle Database Lite version 10.3.0.3.0 to Oracle Database Mobile Server 12c. Therefore, if your current version of Oracle Database Lite is prior to version 10.3.0.3.0, you must first upgrade to Oracle Database Lite 10.3.0.3.0 and apply all patches before upgrading to Oracle Database Mobile Server 12c. For more information on upgrading Oracle Database Lite versions 10.1 and 10.2 to Oracle Database Lite version 10.3, see [Section 5.4, "Upgrade Oracle Database Lite from 10g Release 1 or 2 to 10g Release 3"](#).

5.5.2 Upgrade Mobile Server

Upgrading Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 12c can be performed on an installation in the same ORACLE_HOME or in a separate ORACLE_HOME.

Table 5–14 shows what version of Oracle Database Lite 10.3.0.3.0 and application server installation can be upgraded to Oracle Database Mobile Server 12.1.0

Table 5–14 Upgrading Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 12.1.0

From Oracle Database Lite 10.3.0.3.0	To Oracle Database Mobile Server 12.1.0 Base	Same Oracle Home	Separate Oracle Home
Oracle Database Lite 10.3.0.3.0 Standalone	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c Release 1	Supported	Supported
	GlassFish 3.1.2	Supported	Supported
	Apache TomEE 1.6	Supported	Supported
Oracle Database Lite 10.3.0.3.0 with WebLogic 11g Release 1	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c Release 1	Supported	Supported
	GlassFish 3.1.2	Supported	Supported
	Apache TomEE 1.6	Supported	Supported
Oracle Database Lite 10.3.0.3.0 with Oracle iAS 10.1.3.5.0	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c Release 1	Supported	Supported
	GlassFish 3.1.2	Supported	Supported
	Apache TomEE 1.6	Supported	Supported

Upgrade OID

If you use OID 10.1.2 to manage user information for Oracle Database Lite 10.3.0.3.0, upgrade to OID version 11.1.1.5. Refer to *Oracle Fusion Middleware Upgrade Guide for Oracle Identity Management 11g Release 1 (11.1.1)* for detailed instructions on how to upgrade OID.

General Guideline

1. If you are using Oracle Database Lite 10.3.0.3.0 Mobile Server Standalone, migrate to one of the supported application servers on a separate ORACLE_HOME: WebLogic, GlassFish, Apache TomEE. See [Section 3.3.3, "Certified Application Server Configurations"](#) for details on supported application servers.
2. If you are currently using WebLogic, you can do one of the following:
 - You can continue to use the installed application server as your application server as long as the version is a supported version.
 - If you plan to upgrade Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 12.1.0, and switch to a different application server, you can install Mobile Server either in the same ORACLE_HOME or in a separate ORACLE_HOME. See [Section 5.5.2.1, "Upgrade in the same ORACLE_HOME"](#) and [Section 5.5.2.2, "Upgrade in a separate ORACLE_HOME"](#) for more information.
3. If you have OracleAS installed as the application server, but you want to migrate to WebLogic Server, refer to Oracle WebLogic SmartUpgrade directions on the

following site:

<http://www.oracle.com/technetwork/middleware/downloads/smartupgrade-085160.html>

4. You must apply all patches before upgrading from Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 12c.

See the following sections:

- [Section 5.5.2.1, "Upgrade in the same ORACLE_HOME"](#)
- [Section 5.5.2.2, "Upgrade in a separate ORACLE_HOME"](#)
- [Section 5.5.2.3, "Upgrade Your Mobile Server Repository"](#)
- [Section 5.5.2.4, "Upgrade Your Existing Applications"](#)
- [Section 5.5.2.5, "Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory \(OID\)"](#)

5.5.2.1 Upgrade in the same ORACLE_HOME

Refer to [Table 5–14](#) for the supported cases for upgrading Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 12c in the same ORACLE_HOME.

1. Install Oracle Database Mobile Server 12c in ORACLE_HOME.
 - Launch Oracle Universal Installer. On "Select Installation Type" page, select "Mobile Server".
 - On "Specify Home Details" page, choose the same ORACLE_HOME as your previous installation.
 - Follow the instructions to complete the install.
2. Execute the Repository Wizard to upgrade the repository.
 - If you choose to run Repository Wizard during install, the Repository Wizard launches automatically.
 - Alternatively, you can launch Repository Wizard after installation. Refer to [Section 5.5.2.3, "Upgrade Your Mobile Server Repository"](#).
3. Execute the application upgrade assistant to upgrade user applications to the new repository. For more information, refer to [Section 5.5.2.4, "Upgrade Your Existing Applications"](#).

5.5.2.2 Upgrade in a separate ORACLE_HOME

Refer to [Table 5–14](#) for the supported cases for upgrading Oracle Database Lite 10.3.0.3.0 to Oracle Database Mobile Server 12c in a separate ORACLE_HOME.

1. Install Oracle Database Mobile Server 12c in a new ORACLE_HOME.
 - Launch Oracle Universal Installer. On "Select Installation Type" page, select "Mobile Server".
 - On "Specify Home Details" page, choose a new ORACLE_HOME that is different from your previous installation.
 - Follow the instructions to complete the install.
2. Execute the Repository Wizard to upgrade the repository.
 - If you choose to run Repository Wizard during install, the Repository Wizard launches automatically.

- Alternatively, you can launch the Repository Wizard after installation. Refer to [Section 5.5.2.3, "Upgrade Your Mobile Server Repository"](#).
- 3. Execute the application upgrade assistant to upgrade user applications to the new repository. For more information, refer to [Section 5.5.2.4, "Upgrade Your Existing Applications"](#).
- 4. Uninstall the previous mobile server version from the old ORACLE_HOME.
 - Launch Oracle Universal Installer, click "Deinstall Products".
 - Choose the previous mobile server that you want to uninstall, and click "Remove".

5.5.2.3 Upgrade Your Mobile Server Repository

During installation of Oracle Database Mobile Server, the Repository Wizard detects if Oracle Database Lite 10.3.0.3.0 is installed and initiates the upgrade process, as follows:

Note: If you need to start the Repository Wizard independent of the install process, execute one of the following:

- Windows: %ORACLE_HOME%\Mobile\Server\admin\repwizard.bat
 - Unix: \$ORACLE_HOME/mobile/server/admin/repwizard
-
-

1. Enter the SYSTEM password. Click "Next".
2. Select the schema that you are going to upgrade. Click "Next".
3. Enter the password for the mobile server repository and click **YES** if you want to install the sample applications. Click "Next".
4. A summary screen appears. Click "Next".
5. After the repository is upgraded, click "Finish".

In Oracle Database Mobile Server 12c, the repository location is set to %ORACLE_HOME%\Mobile\Server\admin\repository directory by default, which is different from Oracle Database Lite 10.3.0.3.0.

5.5.2.4 Upgrade Your Existing Applications

You can upgrade all existing applications with the Application Upgrade Assistant. Since the Oracle Database Mobile Server 12c repository is in a new location, the Application Upgrade Assistant moves applications from the old repository location to the new one.

The Application Upgrade Assistant is located at:

- On Windows: %ORACLE_HOME%\Mobile\Server\admin\ApplicationUpgradeAssistant.bat
- On Unix: \$ORACLE_HOME/mobile/server/admin/ApplicationUpgradeAssistant.sh

On the command-line, supply the following:

- Mobile server repository name and password
- The ORACLE_HOME for the old mobile server repository

For example, on a Windows system, where the repository schema name is **mobileadmin**, password is **manager**, the old ORACLE_HOME is **C:\Oracle\ora10gR3**, execute the Application Upgrade Assistant as follows:
ApplicationUpgradeAssistant.bat mobileadmin manager C:\Oracle\ora10gR3

The Application Upgrade Assistant will not upgrade applications for de-supported platforms, as follows:

- Applications deployed on de-supported platforms such as Oracle Lite, Web-to-Go, OC4J and BC4J platforms will not be upgraded.
- The default system folders and default application files for previous releases will not be upgraded, which includes admin-web, msadmin, olsetup, devmgr, META-INF, plugins, setup, and webtogo.
- In Oracle Database Mobile Server 12c, the Mobile Manager application is named 'mobile'. If you deployed an application with the same name in Oracle Database Lite 10.3.0.3.0, the Application Upgrade Assistant will not upgrade it due to a name conflict. You will need to manually change your application name and redeploy it on the new Mobile Server.

5.5.2.5 Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory (OID)

You can use the Oracle Internet Directory (OID), which is part of the Oracle Application Server, for storing and retrieving user information instead of the Mobile Server Repository. To use OID, you must migrate all user information from the existing repository into OID. For more information, refer to Section 4.3.1.6 "Managing OID Users in the Mobile Server" in the *Oracle Database Mobile Server Administration and Deployment Guide*.

5.5.3 Mobile Server Farm Upgrade

If you have multiple mobile servers in a farm that use the same mobile server repository on a back-end Oracle database, perform the following:

1. Shut down all mobile servers before you start the upgrade.
2. Upgrade one of these mobile servers and the mobile server repository.
3. Upgrade each additional mobile server that uses the repository.
4. Restart all mobile servers.

5.5.4 Upgrade Mobile Client

Berkeley DB and SQLite Mobile Clients installed with the Oracle Database Lite Mobile Server version 10.3.0.3.0 can be upgraded to Oracle Database Mobile Server 12c.

[Table 5–15](#) shows what mobile client platforms can be upgraded to the Oracle Database Mobile Server 12c.

Table 5–15 Upgrading Mobile Clients

Client Device Platform	Upgrade Instructions
Berkeley DB or SQLite Mobile Clients on Win32	Run <code>update.exe</code> or <code>msync.exe</code> , which launches <code>update.exe</code> after synchronization completes.
Berkeley DB or SQLite Mobile Clients on Pocket PC 5.0, 6.0, 6.5	Run <code>update.exe</code> or <code>msync.exe</code> , which launches <code>update.exe</code> after synchronization completes.

Table 5–15 (Cont.) Upgrading Mobile Clients

Client Device Platform	Upgrade Instructions
Berkeley DB or SQLite Mobile Clients on Linux	Run <code>update</code> or <code>msync</code> , which launches <code>update</code> after synchronization completes.
SQLite Mobile Client on Android	Run Oracle Update application.
SQLite Mobile Client on Blackberry	De-install the previous client and install a new mobile client

To upgrade mobile client on Android, if server port has changed in server upgrade, modify `SERVER_URL` parameter in [NETWORK] section in `devmgr.ini` before running the Oracle Update application. Usually you can find `devmgr.ini` at `/mnt/sdcard/oracle.mobileclient`.

To upgrade mobile client on Win32, Windows Mobile or Linux platforms, follow the instructions below:

1. Shut down your applications.
2. Check and install available software updates. Ensure that your device is enabled to receive software updates from the Mobile Server. You can either run `update.exe` or `msync.exe`, which automatically launches `update.exe` after synchronization completes.

`Update.exe` might not work if the server port is changed in upgrade. In this case follow the instructions below:

- Write `OSE.FILES=YES` in [DEFAULT] section in `ose.ini`. You can set `OSE.FILES=NO` if you want to disable `OSE.FILES` option after the client upgrade process completes.
 - Launch `msync.exe` and specify the new server URL in 'Server URL' field. Click 'Sync' to perform synchronization. After synchronization completes, `update.exe` launches automatically and is able to connect to Mobile Server with the new server URL. `Update.exe` should find available software updates on the server and display the options for you to install the update.
3. After you update your mobile client, check if `NAME` and `TYPE` parameters are written in [DMC] section in `devmgr.ini`. If they are missing, copy them from the old configuration file `polite.ini`.
 4. If you are upgrading SQLite client, the client upgrade process is completed. If you are upgrading Berkeley DB client, double check if the old Berkeley DB environment files are removed during client upgrade. If not, follow the instructions to finish the upgrade process:
 - Shutdown your applications and `SyncAgent` if it is running.
 - Backup the Berkeley DB databases.
 - Remove environment files `__db.xxx` (xxx stands for 3 digits) in `*.db-journal` folders from where Berkeley DB databases are stored.
 - Restart your application and `SyncAgent` if needed.

See the following section:

- [Section 5.5.4.1, "Upgrade Oracle Mobile Clients from De-Supported Mobile Client Platforms"](#)

5.5.4.1 Upgrade Oracle Mobile Clients from De-Supported Mobile Client Platforms

For Web-to-Go, OC4J, BC4J, and Branch Office Mobile Client platforms, there are no upgrade options since these client device platforms are no longer supported in Oracle Database Mobile Server 12c.

Table 5–16 shows how to migrate de-supported Mobile client platforms to available platforms in Oracle Database Mobile Server 12c.

Note: Do not cancel in the middle of the upgrade process.

Table 5–16 *Upgrading from De-Supported Mobile Client Platforms*

De-Supported Mobile Client Platform	Upgrade to the Available Platform
Berkeley DB or SQLite Mobile Client on Pocket PC 2003	Berkeley DB or SQLite Mobile Client on Pocket PC 6.0

5.6 Upgrade Oracle Database Mobile Server 11g to 12.1.0

The following sections describe the steps for upgrading your software:

- [Section 5.6.1, "Upgrade Mobile Server"](#)
- [Section 5.6.2, "Mobile Server Farm Upgrade"](#)
- [Section 5.6.3, "Upgrade Mobile Client"](#)

5.6.1 Upgrade Mobile Server

Upgrading Oracle Database Mobile Server 11g (11.1.0, 11.2.0 or 11.3.0) to Oracle Database Mobile Server 12.1.0 can be performed on an installation in the same ORACLE_HOME or in a separate ORACLE_HOME. You must apply all patches before upgrading Oracle Database Mobile Server 11g to 12.1.0. Table 5–17 shows which version of Oracle Database Mobile Server 11g and application server installation can be upgraded to Oracle Database Mobile Server 12.1.0.

The following sections describe how to perform the upgrade:

- [Section 5.6.1.1, "Upgrading in the Same ORACLE_HOME"](#)
- [Section 5.6.1.2, "Upgrading in a Separate ORACLE_HOME"](#)
- [Section 5.6.1.3, "Upgrade Your Mobile Server Repository"](#)
- [Section 5.6.1.4, "Upgrade Your Existing Applications"](#)
- [Section 5.6.1.5, "Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory \(OID\)"](#)

Table 5–17 Upgrading Oracle Database Mobile Server 11g to 12.1.0

From Oracle Database Mobile Server 11g	To Oracle Database Mobile Server 12.1.0 Base	Same Oracle Home	Separate Oracle Home
Mobile Server 11.1.0, 11.2.0 or 11.3.0 with Oracle WebLogic Server 11g Release 1,	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c Release 1	Supported	Supported
Mobile Server 11.2.0 or 11.3.0 with Oracle WebLogic Server 12c Release 1	GlassFish 3.1.2	Supported	Supported
	Apache TomEE 1.6	Supported	Supported
Mobile Server 11.1.0, 11.2.0 or 11.3.0 with GlassFish 3.1	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c Release 1	Supported	Supported
	GlassFish 3.1.2	Supported	Supported
	Apache TomEE 1.6	Supported	Supported
Mobile Server 11.2.0, 11.3.0 with Apache TomEE 1.0	Oracle WebLogic Server 11g Release 1, Oracle WebLogic Server 12c Release 1	Supported	Supported
	GlassFish 3.1.2	Supported	Supported
	Apache TomEE 1.6	Supported	Supported

5.6.1.1 Upgrading in the Same ORACLE_HOME

Refer to [Section 5.5.2.1, "Upgrade in the same ORACLE_HOME"](#).

5.6.1.2 Upgrading in a Separate ORACLE_HOME

Refer to [Section 5.5.2.2, "Upgrade in a separate ORACLE_HOME"](#).

5.6.1.3 Upgrade Your Mobile Server Repository

Refer to [Section 5.5.2.3, "Upgrade Your Mobile Server Repository"](#).

5.6.1.4 Upgrade Your Existing Applications

You can upgrade all existing applications with the Application Upgrade Assistant.

The Application Upgrade Assistant is located as follows:

- On Windows: %ORACLE_HOME%\Mobile\Server\admin\ApplicationUpgradeAssistant.bat
- On Unix: \$ORACLE_HOME/mobile/server/admin/ApplicationUpgradeAssistant.sh

On the command-line, supply the following:

- Mobile server repository name and password
- The ORACLE_HOME for the old mobile server repository

For example, on a Windows system, where the repository schema name is **mobileadmin**, password is **manager**, the old ORACLE_HOME is **C:\Oracle\MobileServer11g**, execute the Application Upgrade Assistant as follows:
ApplicationUpgradeAssistant.bat mobileadmin manager C:\Oracle\MobileServer11g

5.6.1.5 Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory (OID)

Refer to [Section 5.5.2.5, "Migrate Your Users from the Mobile Server Repository to Oracle Internet Directory \(OID\)"](#).

5.6.2 Mobile Server Farm Upgrade

Refer to [Section 5.5.3, "Mobile Server Farm Upgrade"](#).

5.6.3 Upgrade Mobile Client

Berkeley DB and SQLite Mobile Clients installed with the Oracle Database Mobile Server 11g can be upgraded to version 12.1.0. [Table 5–18](#) shows what mobile client platforms can be upgraded to the Oracle Database Mobile Server 12.1.0.

Table 5–18 Upgrading Mobile Clients

Client Device Platform	Upgrade Instructions
Berkeley DB or SQLite Mobile Clients on Win32	Run <code>update.exe</code> or <code>msync.exe</code> , which launches <code>update.exe</code> after synchronization completes.
Berkeley DB or SQLite Mobile Clients on Pocket PC 5.0, 6.0, 6.5	Run <code>update.exe</code> or <code>msync.exe</code> , which launches <code>update.exe</code> after synchronization completes.
Berkeley DB or SQLite Mobile Clients on Linux	Run <code>update</code> or <code>msync</code> , which launches <code>update</code> after synchronization completes.
Berkeley DB or SQLite Mobile Client on Android	Run Oracle Update application.
SQLite Mobile Client on Blackberry	De-install the previous client and install the latest version.
Berkeley DB or SQLite Mobile Client on iOS	Develop applications with the latest iOS mobile client package.

- Android Mobile Client Upgrade
 1. Shut down the applications.
 2. If the mobile server port is changed in upgrade, modify the `SERVER_URL` parameter in [NETWORK] section in `devmgr.ini` before running the Oracle Update application. Usually you can find `devmgr.ini` at `/mnt/sdcard/oracle.mobileclient`.
 3. Backup Berkeley DB databases before Berkeley DB mobile client upgrade. For more information, see [Section 5.6.3.1, "Upgrade Berkeley DB Mobile Client"](#).
 4. Run Oracle Update application to upgrade mobile client.
 5. Re-create Berkeley DB databases after Berkeley DB mobile client upgrade. For more information, see [Section 5.6.3.1, "Upgrade Berkeley DB Mobile Client"](#).
- iOS Mobile Client Upgrade
 1. Rebuild applications for Berkeley DB or SQLite mobile client with the 12.1.0 iOS mobile client package. See Chapter 6, "Creating Sync Application for iOS" in *Oracle Database Mobile Server Mobile Client Guide*.
 2. In Berkeley DB iOS mobile client upgrade, you need to backup and re-create Berkeley DB databases. See [Section 5.6.3.1, "Upgrade Berkeley DB Mobile Client"](#).

- Win32, Windows Mobile and Linux Mobile Client Upgrade
 1. Shut down the applications.
 2. Backup Berkeley DB databases before Berkeley DB mobile client upgrade. See [Section 5.6.3.1, "Upgrade Berkeley DB Mobile Client"](#).
 3. Check and install available software updates. Ensure that your device is enabled to receive software updates from the Mobile Server. You can either run `update.exe` or `msync.exe`, which automatically launches `update.exe` after synchronization completes. Refer to *Section 7.6.3, "Initiate Updates of Oracle Database Mobile Server Software from the Client"* in *Oracle Database Mobile Server Administration and Deployment Guide*.

Note: The `update.exe` might not work if the mobile server port has changed in upgrade. In this case follow the instructions below:

- Launch `msync.exe` and specify the new server URL in 'Server URL' field.
- Click 'Sync' to perform synchronization.

After synchronization completes, `update.exe` launches automatically and is able to connect to Mobile Server with the new server URL. `Update.exe` should find available software updates on the server and display the options for you to install the updates.

4. Re-create Berkeley DB databases after Berkeley DB mobile client upgrade. For more information, see:
 - [Section 5.6.3.1, "Upgrade Berkeley DB Mobile Client"](#)

5.6.3.1 Upgrade Berkeley DB Mobile Client

In Oracle Database Mobile Server 12.1.0, Berkeley DB (BDB) libraries are updated to version 6.1. Because the binary format of BDB 6.1 is not compatible with that of the previous BDB releases shipped with Oracle Database Mobile Server 11g, a direct upgrade of BDB databases would not work. Hence, there are two options to upgrade Berkeley DB databases:

- Backup and dump Berkeley DB databases before upgrade and re-create Berkeley DB 6.1 databases after upgrade.

For this option, follow the steps below:

1. Dump old Berkeley DB databases (*.db) stored under data directory before Berkeley DB Mobile Client upgrade. For example, if you have 4 databases under data directory, use `.dump` command of old `dbsql` to dump the databases into SQL text files as follows:

```
prompt>dbsql.exe ..\bdb\data\JOHN\test.db .dump > test.sql
prompt>dbsql.exe ..\bdb\data\JOHN\default.db .dump > default.sql
prompt>dbsql.exe ..\bdb\data\JOHN\OSE$test.db .dump > OSE$test.sql
prompt>dbsql.exe ..\bdb\data\JOHN\OSE$default.db .dump > OSE$default.sql
```

You may run out of space of the mobile device where Berkeley DB Mobile Client is installed. In this case, you can copy old Berkeley DB databases, `dbsql` tool and Berkeley DB libraries to a separate system (maybe call it dump system for example) and do Step 1 in dump system.

2. Remove all the files under data directory, including *.db and *.db-journal. In this example, remove all files under bdb\data\JOHN.
3. Upgrade Mobile Server and Berkeley DB Mobile Client to 12.1.0.
4. Re-create Berkeley DB databases from the SQL text files with dbsql 6.1.

```
prompt>dbsql.exe ..\bdb\data\JOHN\test.db < test.sql
prompt>dbsql.exe ..\bdb\data\JOHN\default.db < default.sql
prompt>dbsql.exe ..\bdb\data\JOHN\OSE$test.db < OSE$test.sql
prompt>dbsql.exe ..\bdb\data\JOHN\OSE$default.db < OSE$default.sql
```

If you do Step 1 and Step 4 in a dump system, you need to copy the new Berkeley DB databases back to the data directory of Berkeley DB Mobile Client in the mobile device.

- Use Mobile Server repository for data backup.

For this option, follow the steps below:

1. Before the upgrade, synchronize Berkeley DB Mobile Client with Mobile Server to ensure that all client data is saved in the Mobile Server repository.
2. Uninstall old Berkeley DB Mobile Client. You can either uninstall the Mobile Client manually, or send the 'uninstall' command from Mobile Server to uninstall Mobile Client on Win32, Windows Mobile and Linux platforms automatically.
3. Upgrade Mobile Server to 12.1.0 and install Berkeley DB Mobile Client 12.1.0.
4. Do a sync with the new Berkeley DB Mobile Client. Berkeley DB 6.1 databases will be created under data directory.

Migrating from Oracle Lite Client Database to Berkeley DB

The Oracle Lite Client Database is no longer supported as of Oracle Database Mobile Server 11g. Oracle recommends replacing the Oracle Lite Client Database (OLite) with Oracle Berkeley DB: a fast, highly scalable, ACID compliant database. For those users using the Oracle Lite Client Database to synchronize with the backend to an Oracle Database through Oracle Database Mobile Server, the following [Section 6.1, "Migration Guideline"](#) provides helpful information and steps to migrate from OLite to Berkeley DB.

6.1 Migration Guideline

The recommended migration paths for the common OLite use cases are:

1. If you deploy OLite on mobile devices, and the Mobile Server is used to synchronize data between mobile devices and an Oracle database, then it is recommended to replace OLite with Berkeley DB.
2. If you are using Web-to-Go on the client side to run JAVA Servlets, and OLite as the client data store, you need to replace Web-to-Go with a Servlet container and replace OLite with Berkeley DB.
3. If you are using Branch Office, which consists of OLite and a Multi-User Listener as a database service for multiple users and applications, there is no identical functionality in Oracle Database Mobile Server 11g. Contact Oracle sales to design an appropriate solution.

In scenarios 1 and 2 above, you may want to migrate the data stored in OLite to Berkeley DB and modify your applications because of the differences in APIs, SQL syntax, database drivers, etc.

The following sections provide you step by step instructions on how to migrate client side application data from OLite to Berkeley DB.

- [Section 6.1.1, "Backup OLite Database"](#)
- [Section 6.1.2, "Upgrade Oracle Lite Mobile Client"](#)
- [Section 6.1.3, "Export Oracle Lite Database"](#)
- [Section 6.1.4, "Migrate OLite Publication"](#)
- [Section 6.1.5, "Upgrade Oracle Database Lite Mobile Server 10.3.0.3.0 to Oracle Database Mobile Server 12c"](#)
- [Section 6.1.6, "Create Berkeley DB-based Publication and Application"](#)

- [Section 6.1.7, "Install Berkeley DB Mobile Client"](#)
- [Section 6.1.8, "Import Data to Berkeley DB"](#)

After migration, you can synchronize your application data in the backend Oracle Database through Oracle Database Mobile Server.

6.1.1 Backup OLite Database

Before you start migration, backup your OLite database on your mobile devices. You can backup the OLite database either by using the **backupdb** utility or by copying the database files to a safe location. Before any files can be copied, stop all applications that access the OLite database and shut down the multi-user service if it is running. Once completed, execute the **backupdb** utility. To backup OLite database, use the syntax:

```
BACKUPDB DSN DBName backup_filename [DB_password]
```

Refer to *Oracle Database Lite Oracle Lite Client Guide Release 10.3* for the usages of backupdb utility.

6.1.2 Upgrade Oracle Lite Mobile Client

Download and apply the migration patch 17405964 for Oracle Database Lite Mobile Server 10.3.0.3.0 to support exporting OLite database. After you apply the patch, restart the Mobile Server and execute "Update" from Oracle Lite Mobile Client.

A new version of Oracle Lite Mobile Client will be downloaded and installed which enables you to export OLite database.

6.1.3 Export Oracle Lite Database

The newly installed Oracle Lite Mobile Client supports exporting OLite database to binary files using File-based Sync. You can either export data in graphical mode using msync or programmatically using OSE Synchronization API. The steps for exporting data using msync are as follows:

1. Enable OLite data export

Modify polite.ini, write the following in [SYNC] section:

```
OSE_EXPORT_PLUGIN=explite
```

This step could be automated by adding <ini> section to the patch file:

- a. In the patch files:

```
<ORACLE_
HOME>/mobile/server/admin/repository/setup/dmc/[win32 | linux/x86 |
wince/ppc60]/patch_[win32 | linux-x86 | ppc60]_bug#.inf
```

After

```
<action msg_i='$FILE_I$' msg='$FILE_U$'>ini</action>
```

Add

```
<action msg_i='$INI_I$' msg='$INI_U$'>ini</action>
```

Then, add <ini> element after </file>

```
<ini>
```

```
<item name='POLITE.INI' section='SYNC'>
```

```
<item name='OSE_EXPORT_PLUGIN'>explite</item>
</item>
</ini>
```

- b. The same should be done for patch files in the


```
<ORACLE_HOME>/mobile_
oc4j/j2ee/mobileserver/applications/mobileserver/setup_HOME
```

2. Export OLite data

- a. Export data to the default location. Launch msync. The "Server URL" field should show "N/A(Export Mode)". Click "Sync" button to export data. By default it will export data into the following file:

```
$(USER_DIR)\ose_exports\ose_export_$(SYNC_USER).bin
```

USER_DIR is the user home directory on the machine; SYNC_USER is the sync user name. For example: C:\Users\Jack\ose_exports\ose_export_jack.bin

- b. Export data to a customized location

You can also customize the export file name and location as follows:

Launch msync, go to "Tools" - "File Sync Options" menu, check the "File Based Sync", set the mode to "Send" and enter the file location in the "File" field.

Click "Sync" to export data. (Do not click "Apply" button because there is a known issue with the msync UI that if you check "File Based Sync" option and click "Apply" button, msync UI will crash.) Then after export is done, you will get the exported data in the location you specified.

- c. After the export is done, you should see the following message in the msync result window:

```
"The database data has been exported. Please proceed with client upgrade."
```

Now you can uninstall Oracle Lite Mobile Client.

If you want to export OLite data programmatically using OSE Synchronization API, refer to Oracle® Database Mobile Server Developer's Guide on how to use File-based Sync.

6.1.4 Migrate OLite Publication

The existing OLite publications require migration to support Berkeley DB clients. You can migrate the OLite publications before upgrading 10.3.0.3.0 to 12c by running the migration tool MigrateOlitePublication.bat (or MigrateOlitePublication.sh) located at <ORACLE_HOME>/mobile/server/bin to migrate your OLite publications.

You can also write a Java program to migrate the OLite publications by a Consolidator Java API call: addBDBSQLiteDevFlg(String name).

String name is the OLite publication name.

6.1.5 Upgrade Oracle Database Lite Mobile Server 10.3.0.3.0 to Oracle Database Mobile Server 12c

Refer to Oracle Database Mobile Server Documentation Release 12.1.0 on how to upgrade the Mobile Server.

6.1.6 Create Berkeley DB-based Publication and Application

Creation of a new Berkeley DB publication is only required if the OLite publication has not been migrated by the method described in [Section 6.1.4, "Migrate OLite Publication"](#). You can either use Mobile Development Kit (MDK) or write a program using Consolidator Java API to create Berkeley DB publications if needed.

Create a new application or modify the existing application and deploy it to the Mobile Server.

1. Create or modify the application

There are some differences between OLite and Berkeley DB that you need to pay attention to:

- OLite supports Oracle SQL extensions and Berkeley DB does not.
- OLite and Berkeley DB both support SQL-92, but in some cases, modifications will be required.
- OLite and Berkeley DB both support ADO.NET, ODBC, JDBC APIs, but the drivers and usages of these APIs are different.
- Berkeley DB does not support OKAPI, SODA and SODASQL.

Refer to Berkeley DB documentation on how to write applications with Berkeley DB.

2. Deploy the application to Mobile Server

Deploy your application to Mobile Server either using Packaging Wizard or programmatically using Mobile Server APIs. You have two choices:

- Delete the old OLite application and publish the new Berkeley DB application, then re-assign existing users to the new application.
- Modify the existing OLite application. This approach does not require to re-assign existing users to the modified application

6.1.7 Install Berkeley DB Mobile Client

Download Berkeley DB Mobile Client from Oracle Database Mobile Server 12c to your mobile device and install it.

6.1.8 Import Data to Berkeley DB

After installing Berkeley DB Mobile Client, you are ready to import the data that you exported from OLite database to Berkeley DB.

1. Import data with default option

After you install Berkeley DB Mobile Client, if there is an export .bin file in the default location ($$(USER_DIR)\ose_exports\ose_export_$(SYNC_USER).bin$), data migration will happen automatically. The first sync will be setup sync and the next sync will import sync.

2. Customizing data import

Alternatively, if you want the export file to be in a different (non-default) location and name, follow the instructions:

- a. Put the following in [DEFAULT] section in ose.ini to enable setup sync:

```
OSE.SETUP_SYNC=TRUE
```

- b. Launch msync, click “Sync” button to start setup sync. You will see the following message in msync result window after setup sync is done:

"The client databases have been set up. Please sync again to import the data from the binary file generated during the export stage. If you would like to specify custom location for this file, please use the File sync options."

- c. The next sync will automatically be import sync.

Go to “Tools” -> “File Sync Options” menu, check the “File Based Sync”, set the mode to “Receive” and enter the file location in the “File” field. Click “Sync” to start import sync and you will see sync progress dialog. After import sync is done the following message will be displayed:

"The client data has been imported. This concludes the client upgrade."

Now data migration is done.

If you want to import OLite data to Berkeley DB programmatically using OSE Synchronization API, refer to *Oracle Database Mobile Server Developer's Guide* on how to use File-based Sync.

A

accounts
 UNIX, 3-9
application
 mobile, 1-1
 remote, 1-1
architecture
 mobile, 1-1
 benefits, 1-1

C

concepts
 mobile server, 1-1

D

data file
 specify location, 3-4
 specifying location, 3-13
db_create_file_dest parameter, 3-13
DHCP
 install, 4-1, 4-2
DISPLAY property, 3-8

E

enterprise
 remote design, 1-2
environment variables
 UNIX, 3-6

F

farm
 creating, 4-28
 install, 4-28

G

groups
 UNIX, 3-9

H

hardware requirements

Windows, 3-6
high availability
 installation, 4-28

I

install, 4-1
 DHCP, 4-1, 4-2
 instructions, 4-4
 log files, 4-20
 Mobile Development Kit, 4-21
 mobile server, 4-5
 multiple servers, 4-28
 multiple languages, 4-1
 Oracle software requirements, 3-4
 repository, 4-6
 system requirements, 3-1, 3-5, 3-6

K

kernel parameters
 configuring, 3-10

L

language
 multiple language install, 4-1
LD_LIBRARY_PATH, 3-7
LIBPATH, 3-7
Linux
 pre-install requirements, 3-9
load balancing, 4-28

M

middle-tier
 Oracle WebLogic Server, 3-4
mobile
 application, 1-1
 architecture, 1-1
 benefits, 1-1
Mobile Development Kit
 install, 4-21
mobile server
 concepts, 1-1
 install, 4-5

- introduction, 1-1
- load balancing, 4-28
- multiple servers, 4-28
- testing install, 4-31
- uninstall, 4-32

mobile server repository, see repository

mobile.ora

- configuring reverse proxy, 4-28

O

OID

- supported version, 3-4

Oracle database

- data file

 - specify location, 3-4

Oracle Database Mobile Server, see mobile server

Oracle WebLogic Server

- install requirements, 3-4

ORACLE_HOME, 3-7

P

PATH, 3-7

performance

- modify tablespace layout, 3-13

R

remote

- application, 1-1

repository

- install, 4-6
- schema name, 4-13

Repository Wizard

- install, 4-9

reverse_proxy parameter, 4-28

S

shell limits

- UNIX, 3-10

system requirements, 3-1, 3-5, 3-6

- hardware, 3-6
- Oracle software, 3-4

T

tablespace

- layout, 3-13
- location, 3-13
- name, 3-13

TMP, 3-8

TMPDIR, 3-8

U

uninstall

- mobile server, 4-32

UNIX

- creating accounts, 3-9

- creating groups, 3-9
- environment variables, 3-6
- kernel parameters, 3-10
- shell limits, 3-10
- system configuration, 3-6
- system requirements, 3-6

W

Windows

- hardware requirements, 3-6
- system requirements, 3-5