

# Oracle® Database Mobile Server

Release Notes

Release 11.1.0

E22675-02

September 2011

---

The information in this release note pertains to items that did not make it into each book. The information is organized by the book to which it pertains, as follows:

- [Section 1, "Advance Notice"](#)
- [Section 2, "New Features for Oracle Database Mobile Server 11g \(11.1.0.0\)"](#)
- [Section 3, "Known Problems and Limitations"](#)
- [Section 4, "Documentation Accessibility"](#)

## 1 Advance Notice

This section lists deprecated, removed, and desupported items.

- The Oracle Lite Mobile Client is desupported in this release. The Sync Control APIs for the Oracle Lite Mobile Client are still provided in this guide, but will be removed when the Oracle Lite Mobile Client is fully deprecated.

Supported mobile clients include the Berkeley DB and SQLite Mobile Clients.

- Formerly this product shipped with two built-in servlet containers; OC4J on the server side, and Web-to-Go on the client side. However, as of Oracle Database Mobile Server 11g does not ship with a built-in Web application server, neither on the server nor the client side. Users will need to install a supported application server before installing the Oracle Database Mobile Server or mobile clients. On the server side, Oracle WebLogic Server (WLS), Glassfish, and OracleAS are supported. On the client side, the users can choose any servlet container that suits their needs.

## 2 New Features for Oracle Database Mobile Server 11g (11.1.0.0)

- The Oracle Database Mobile Server 11g supports synchronization from only two mobile clients, the Berkeley DB Mobile Client and the SQLite Mobile Client, to a back-end Oracle database.
- Automatic synchronization is supported for the Android platform.
- Device management is supported for the Android platform.
- Support for Berkeley DB encryption.
- Load balancing is now available through Oracle WebLogic Server Clusters or the Glassfish Cluster Server Instance instead of WebCache.

- Because of the removal of Web-to-Go, OC4J, and an installed application server, several of the configuration files, configuration parameters and utility names have been eliminated or modified. Refer to the documentation for specific details on each of these changes.
- There are new Java, native, and ADO.Net client synchronization and Sync Agent APIs. There is also a new pure Java synchronization client.

### 3 Known Problems and Limitations

This section contains known problems and limitations in the following Oracle Database Mobile Server components:

- [Section 3.1, "Installing or Upgrading Oracle Database Mobile Server"](#)
- [Section 3.2, "Mobile Client"](#)
- [Section 3.3, "Mobile Server"](#)
- [Section 3.4, "MDW and Packaging Wizard"](#)
- [Section 3.5, "Mobile Development Kit"](#)

#### 3.1 Installing or Upgrading Oracle Database Mobile Server

- You must check the log files after the Oracle Database Mobile Server install, which is the only way to see if the install succeeded or failed. The details about any failures are documented in the log file.
- During upgrade, all existing schemas and repositories, such as `mobileadmin`, are automatically upgraded. You should first back up any schema or repository before performing the upgrade to save copies of the original files.
- Prior to running `upgrade.exe` to upgrade from Oracle Database Lite 10.3 to Oracle Database Mobile Server 11g, ensure that `OSE.FILES=YES` is set in the `ose.ini` file. You can do this manually or a patch will be available to take care of this change.
- On Windows Mobile, if the reserve proxy is set up with authentication, the initial synchronization may fail. If it fails with an `HTTP 401` error, run synchronization a second time and it will succeed. After the initial failure, Oracle Database Mobile Server sets `HTTP_AUTH=TRUE` in the `devmgr.ini` file.
- When using the Repository Wizard, users can deploy the mobile server on the Web Application Server either as a new domain or as an existing domain. The steps for deploying each of these is slightly different.

To deploy the mobile server on a new domain, perform the following:

1. Choose **Yes** for the **Create a new domain for the Mobile Server** option.
2. Specify a domain name. The WebLogic domain directory is created under `%ORACLE_HOME%\..\user_projects\domains`; the Glassfish domain directory is created under `%ORACLE_HOME%\domains`.
3. Specify an admin user name. For WebLogic, the admin user name is automatically set to `weblogic` and you cannot modify it.

4. Specify a server port for the WebLogic domain, specify both the admin port and server port for the Glassfish domain. Make sure the ports you specified are free at the time of deployment.
5. Specify a password for the admin user. WebLogic 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.

To deploy the mobile server on an existing domain, perform the following:

1. Choose **No** for the **Create a new domain for the Mobile Server** option.
2. Specify an existing domain name.
3. Specify the domain directory for the existing domain. For WebLogic, set the domain directory to the folder where your domain files are stored; for Glassfish, set the domain directory to the parent of the folder where your domain files are stored.
4. Specify the admin user name for that domain.
5. Specify the server port for the WebLogic domain, specify both the admin port and server port for the Glassfish domain.
6. Specify the password for the admin user.

### 3.2 Mobile Client

- On Linux platforms, if a synchronization is initiated from the Device Manager window, a new synchronization window is created, which does not close properly after synchronization is finished. Locate the process id of the automatic synchronization process and manually kill it with the `kill -9` command.
- On Linux, the mobile client uses in-memory mapped files in the `$MOBILE_HOME/01IPC` directory. When there is abnormal behavior with the `syncagent`, `msync` or `dmagent` utilities, these files get corrupted. If these utilities start with the original corrupted files, the programs will not work correctly. This is a known issue for the mobile client on the Linux platform. The workaround is to remove `$MOBILE_HOME\01IPC` directory before starting any of these utilities.
- Do not install MDW and the mobile client on the same platform.
- If you reuse a SQLite Android client in the mobile server, the original client may not uninstall correctly. The problem manifests by not showing updates. In this case, you can delete the device from the mobile server once they are uninstalled through Mobile Devices tab -> Select the check box for the device -> Delete -> Confirm deletion. Ensure that the original client is removed from the mobile server.
- When synchronization executes, it creates a folder to store the Berkeley DB database. The default location is `mobileclient\bdb\data\<db>`. If the `mobileclient\bdb\data\` folder does not exist, an `oseException(-12011)` error is thrown. You must create the `mobileclient\bdb\data\` directory.
- On Blackberry devices, SQLite allows only one connection to a given database at any given time. Opening another connection to the database

throws the `DatabaseIOException` ("File system error (12)") error. Because of this limitation, automatic synchronization is not supported on Blackberry devices. If shared connections are not used, the `OSESession` object lifetime is limited to 1 synchronization event only. Thus, after the synchronization call, close the session whether the synchronization was successful or not.

### 3.3 Mobile Server

- In the Mobile Manager, on the page Mobile Devices -> Administration -> Command Management page, there are several occurrences of the word Olite. These can be ignored and will be removed in a future patch release.
- When you log into the Mobile Manager, the port number for the Mobile Server deployed in Glassfish application server mode might not be displayed correctly. However, this does not affect Mobile server functionality.
- Ensure that the Glassfish domain you specified does not contain any corrupted applications that may cause failures in starting the domain. The deployment process needs to start the domain before deploying the mobile server on your Web application server. Any failures in starting the domain will cause a subsequent application deployment to fail.
- The WSH command line tool used to execute batch commands on the mobile server does not work correctly on 64-bit Windows platforms. The workaround is to provide the equivalent Java commands for WSH, as follows:

1. Include `webtogo.jar` in the classpath, as shown below:

```
set CLASSPATH=%MOBILE_HOME%\Mobile\Server\bin\webtogo.jar;%CLASSPATH%
```

2. Include JDK 1.6 in the PATH, as shown below:

```
set PATH=%JAVA_HOME%\bin;%PATH%
```

3. Replace the `wsh` command with `java oracle.lite.web.Shell` and use the `java oracle.lite.web.Shell [-c | -l | -a | -ap ] command` instead of the `wsh [-c | -l | -a | -ap ] command`.

- The Repository Wizard silent install mode does not support application deployment. After running the Repository Wizard in silent mode, you must manually deploy the mobile server application on your Web application server. Refer to your application server documentation for specific application deployment instructions.

The application directory that needs to be deployed is `%MOBILE_HOME%\Mobile\Server\admin\repository\mobile`.

After a successful deployment, add the following properties to `%MOBILE_HOME%\Mobile\Server\bin\mobile.ora` under the `[MOBILE]` section:

```
[MOBILE]
PORT=<server_port>
DOMAIN=<domain_name>
DOMAIN_CONFIG=<domain_config_file>
```

For example:

If you deploy the mobile server on WebLogic in the `mobileserver` domain with port 7001 and your WebLogic `user_projects` directory is `C:\Oracle\Middleware\user_projects`, add the following properties:

```
PORT=7001
DOMAIN=mobileserver
DOMAIN_CONFIG=C:\Oracle\Middleware\user_
projects\domains\mobileserver\config\config.xml
```

If you deploy the mobile server on a Glassfish server installed in the `C:\glassfish3` directory and in the `mobileserver` domain with HTTP listener on port 8080, add these domain specific parameters:

```
PORT=8080
DOMAIN=mobileserver
DOMAIN_
CONFIG=C:\glassfish3\glassfish\domains\mobileserver\config\domain.xml
```

To deploy OracleAS 10.1.3 with Mobile Server 11g, you must configure it manually, as follows:

1. Install OracleAS 10.1.3 and apply the 10.1.3.5.0 patch on top of OracleAS 10.1.3.
2. Stop the OracleAS server with the `opmctl stopall` command, which is in the `<ORACLE_HOME>/opmn/bin` directory.
3. Replace JDK 1.5. in `<ORACLE_HOME>/jdk` with JDK1.6.
4. Copy `ojdbc6.jar` to the `<ORACLE_HOME>/jdbc/lib` directory.
5. Start the OracleAS server with the `opmctl startall` command.
6. Install Oracle Database Mobile Server 11g on top of OracleAS.
7. Select **No** when prompted for running the repository wizard.
8. Create the `mobile.ear` file, as follows:

- a. Set the `JAVA_HOME` and `PATH`.

```
set JAVA_HOME=C:\jdk1.6.0
set PATH=%JAVA_HOME%\bin;%PATH%
```

- b. Change to the `%MOBILE_HOME%\Mobile\Server\admin\repository\mobile` directory, and use the following `jar` command to create the `mobile.ear` file:

```
jar -cvf mobile.ear
```

9. Create the `console.war` file, as follows:

- a. Set the `JAVA_HOME` and `PATH`.

```
set JAVA_HOME=C:\jdk1.6.0
set PATH=%JAVA_HOME%\bin;%PATH%
```

- b. Change to the `%MOBILE_HOME%\Mobile\Server\admin\repository\mobile\console` directory, and use the following `jar` command to create the `console.war` file:

```
jar -cvf console.war
```

10. Once the mobile server is installed, copy `mobileserver.ear` to the `<ORACLE_HOME>\Mobile\Server\admin` directory.
11. Copy `console.war` to the `<ORACLE_HOME>\Mobile\Server\admin\repository\admin-web` directory.
12. Start the `repwizard`. Once the `repwizard` is completed, stop OracleAS and the mobile server with the `opmctl stopall` command.
13. Edit the `web.xml` present in the following location:

```
<ORACLE_HOME>/j2ee/mobileserver/applications/mobileserver/server/WEB-INF
```

Replace the `$ORACLE_HOME` string with the actual location. For example, the following configures the Oracle home location as `C:\ias1010300`.

```
<init-param>
<param-name>webtogo.ora</param-name>
<param-value>C:\ias1010300\mobile\server\bin\mobile.ora</param-value>
</init-param>
```

14. Start the mobile server and the OracleAS server with the `opmctl startall` command.
15. Access the mobile server page with the following URL:

```
http://<hostname>:<port>/mobile
```

### 3.4 MDW and Packaging Wizard

- On Linux, when trying to close a project, the `dmagent` does not always exit properly.
- Before starting to test a publication, make sure the `dmagent` and `syncagent` processes are not up or running. If a previous test of a publication failed, the `dmagent` and `syncagent` may not have shut down correctly and will cause a problem when you execute another publication test.
- On Windows Mobile, using remote synchronization with MDW over RAPI protocol may produce the following error:

```
Unable to send command to the device! RAPI: Command failed (2)
```

If you get this error, check the `devmgr.ini` file and verify that the `NAME` and `TYPE` parameters in the `[DMC]` section are set correctly, as demonstrated below:

```
[DMC]
NAME=BDB PPC60 ARMV4I;US
TYPE=WINCE_ARMV4I_US_BDB_60
```

```
[NETWORK]
ADDRESS=RAPI
PROVIDER=RAPI
SERVER_URL=<URL NAME>
```

- When using MDW and the packaging wizard to publish Java applications, the user should always select the SQLite database, even when using Berkeley DB. This is a result of the common database interface for both SQLite and

Berkeley DB, MDW and the packaging wizard are components of MDK, the Mobile Development Kit.

- You will receive an error if you attempt to publish an application using the packaging wizard on 64-bit operating system without an installed 32-bit JDK. The packaging wizard requires a 32-bit JDK, even if used on a 64-bit operating system. Alternatively, you can create a JAR file with MDW and packaging wizard on another system that has the 32-bit JDK and then upload the JAR file to the mobile manager.
- A JDK 1.6 bug that prevents the `JPasswordField` from receiving keyboard input, which prevents the password from being entered into MDW on Linux platforms. The workaround is to kill any Smart Common Input Method (`scim`) processes before running MDW.
- On the Linux platform, if users terminate the shell command that launches the `oramdw.sh` script, they may have a JVM process with `oracle/lite/mada/core/MADAMainFrame` running in their system. Any further attempts to run MDW with `oramdw.sh` fails. Kill the JVM process and re-execute `oramdw.sh`.
- After running MDW on the Windows system, the `ocmd.exe` processes continue to execute. While these processes do not block MDW functionality, they should be killed prior to the uninstall process. Failure to remove them prevents the MDK folder from being removed during uninstall.
- If MDW is run with different users on the same mobile server, problems can exist if the users are not switched correctly. On Windows, kill all client-side processes including `msync.exe`, `syncagent.exe`, `autosync.exe`, `dmagent.exe`, `update.exe`, `ocmd.exe`, and `setup.exe`. Then remove the `%MOBILE_HOME%\Mobile\Sdk\bin\oseconf` folder before starting MDW. On Linux, the user will have to reboot the system before starting MDW.
- The following exceptions may appear on the MDW console:
  - `WIZARD_FOCUS_SUSTAIN`, `java.lang.RuntimeException` that is thrown from the `AWT-EventQueue-0` thread. Thrown when the user clicks the Finish button at the end of the wizard. You can ignore this error.
  - `java.lang.Exception: 500 Internal Server Err at oracle.lite.web.xml.JupDevApp.publishDef` is thrown when publishing a publication if the an incorrect user name, password or URL for the repository is provided. Re-publish with the user name, password or URL for the repository.
  - `NullPointerException` is thrown when opening an invalid project for the repository. You should drop the project and recreate a valid one.

### 3.5 Mobile Development Kit

- The MDK uses a 32-bit JVM. If you are running on a system with a 64-bit JVM installed, you must also install a 32-bit JVM where the `JAVA_HOME` and `PATH` environment variables point to the 32-bit JVM, in order for the MDK to work as expected.
- Offline instantiation is currently only supported on Windows platforms.

- When creating an offline instantiation package, the client database will be stored under the MDK `%MOBILE_HOME%\Mobile\Sdk\data` directory. Create this folder ahead of time if it does not already exist. If the directory is missing, the following message is generated:

```
Cannot find database root in {0}
```

## 4 Documentation Accessibility

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

### Access to Oracle Support

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

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

---

Oracle Database Mobile Server Release Notes, Release 11.1.0  
E22675-02

Copyright © 1997, 2011, 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, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

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

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

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