Oracle® Database Lite

Release Notes

10*g* (10.0.0)

Part No. B13814-01

June 2004

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:

1 General Subjects

The following subjects apply, in general, to the entire product:

1.1 Translation for Mobile Manager Help Screens

In this release, the help screens for Mobile Manager is only available in English. It is not currently translated into other languages. However, the help will be translated in the next release.

1.2 Using webtogo.exe to Start Mobile Server

You should no longer use the `webtogo.exe` to launch the standalone version of the Mobile Server. Instead, launch the Mobile Server by launching OC4J standalone. See the *Oracle Database Lite Getting Started Guide* for more information.

2 Modifications to the Getting Started Guide

The following subjects detail modifications that should be in the *Oracle Database Lite Getting Started Guide*.

2.1 Using Mobile Manager When Installed On Oracle9*i* Application Server Version 9.0.2

When you have installed Oracle Database Lite on top of Oracle 9*i*AS **version 9.0.2 ONLY**, then you have to manually modify the OC4J `application.xml` file to enable the Mobile Manager. Add the following `library` paths in this file before the other `library` elements, as follows:

```
<library path="../../../mobile/server/bin" />
<library path="../../../mobile/classes" />
```

After modification, use `DCMCTL` to notify OC4J of the changes and restart OC4J. The `application.xml` file is located in the *`ORACLE_HOME`*`\j2ee\home\config` directory. See the Oracle Application Server documentation for more information on `DCMCTL`.

2.2 Installing Multiple Languages on a Single Machine

With Oracle Database Lite 10*g* 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.

2.3 Mobile Server on a DHCP Server is Not Supported

Mobile Server can only be installed on a server with a static IP address; thus, the Mobile Server does not function correctly if installed on a DHCP server.

2.4 What Version of Oracle9*i* Application Server or Oracle Application Server Can the Mobile Repository Interact With?

The Mobile Repository supports running against Oracle9*i* Application Server (9.0.2), Oracle9*i* Application Server Release 2 (9.0.3) or Oracle Application Server 10*g*.

2.5 Configuring for Default Sync When Installing the Client

In the default configuration, all Mobile Clients do not automatically Sync after you install the client. However, you can modify your configuration to automatically sync each client after it is installed, as follows:

1. Logon to the Mobile Server as an Administrator and launch the Mobile Manager tool.
2. Click on Mobile Devices, followed by Administration.
3. Click on Command Management.
4. Edit the Command Device Info (Retrieve device information).
5. Insert 'Synchronize' as a Selected Command and click Apply to accept the changes.

2.6 How Do You Encrypt All Databases for the Initial Sync?

In the default server configuration, all Mobile Clients do not automatically encrypt the local snapshots after you complete the initial sync. However, you can modify your configuration to automatically encrypt all local snapshots with the synchronization user password after the initial sync completes. You can configure for this option either on the client or on the server, as follows:

2.6.1 Configuring on the Local Client for Automatic Encryption of Local Snapshots

On the local client, you can configure for automatic encryption of the local snapshots after initial synchronization by modifying the `polite.ini/polite.txt` file with the following parameter:

```
[SYNC]
ENCRYPDB=1
```

For more information on modifying the `ENCRYPDB` parameter in the `polite.ini/polite.txt` file, see Appendix J, "The Consolidator Client API (OCAPI)", in the *Oracle Database Lite Administration and Deployment Guide*.

2.6.2 Configuring on the Server for Automatic Encryption of Local Snapshots

On the server, you can configure for automatic encryption of the local snapshots after initial synchronization by performing the following:

1. Logon to the Mobile Server as an Administrator and launch the Mobile Manager tool.
2. Click on Mobile Devices, followed by Administration.
3. Click on Command Management.
4. Click **Create Command**.
5. Create the following new Command:

   ```
   Name: EncryptDB
   Command: updt_conf.otl
   Description: Encrypt Database
   ```

6. Edit the newly created command `EncryptDB`, as follows:

   ```
   Command: updt_conf?app=polite/sync&key=ENCRYPDB&val=1
   ```

7. Apply the changes.
8. Edit the `DeviceInfo` Command. Insert the new Command `EncryptDB` and click **OK**.

2.7 JRE Requirement for a Branch Office Client

The minimum requirement for the Branch Office Client installation is `JRE1.3.1_03` or greater. If you use a previous version than `JRE1.3.1_03`, then the Branch Office Manager Tool throws the following exception and hangs. It has been identified as a Java bug in `JRE1.3.1_01`.

```
java.lang.IllegalArgumentException: Signal already used by VM: SIGINT
  at sun.misc.Signal.handle(Unknown Source)
  at java.lang.Terminator.setup(Unknown Source)
  at java.lang.Shutdown.add(Unknown Source)
  at java.lang.Runtime.addShutdownHook(Unknown Source)
  at sun.awt.windows.WToolkit.run(Unknown Source)
  at java.lang.Thread.run(Unknown Source)
```

2.8 Enabling Branch Office on Windows XP Service Pack 2

When you install Windows XP Service Pack 2, the Internet Connection Firewall (ICF) defaults to ON. In order for the Branch Office Server to work properly, you either need to turn the ICF OFF or enable port 100 within the ICF. To enable port 100, go to the Windows Firewall control on your Windows machine. Select the Exception tab. Click **Add Port**. Add port 100 with any name.

3 Modifications to the Administration and Deployment Guide

The following subjects detail modifications that should be in the *Oracle Database Lite Administration and Deployment Guide*.

3.1 Configuring OC4J to Handle Multibyte Characters in Online Web Applications

If you have an application that uses multibyte characters and runs in online mode, you need to configure the `default-charset` element to `Shift_JIS` in the OC4J `global-web-application.xml` file to allow multibyte characters, as follows:

```
<orion-web-app
   deployment-version="1.0.2.2"
   jsp-cache-directory="./persistence"
   temporary-directory="./temp"
   servlet-webdir="/servlet/"
   default-charset="Shift_JIS">
</orion-web-app>
```

The `global-web-application.xml` file can be found in the *ORACLE_HOME*`/mobile_oc4j/j2ee/home/config` directory. For more information on the elements in the `global-web-application.xml` file, see the *Oracle Application Server Containers for J2EE Servlet Guide*.

3.2 Custom Workspace Is Only Supported In Offline Mode

Custom workspace is only supported in offline mode; thus, you cannot go to online mode from this workspace.

3.3 Troubleshooting An Out of Memory Error

If you receive an `Out of Memory` error, then you must allocate more memory to the JVM when starting Mobile Server. In addition, allocating more memory increases your performance. The memory requirements of the Mobile Server can vary dramatically across installations. The amount of memory that your applications use—for example, when propagating data, and so on—can exceed the amount of memory available in the JVM. The administrator can allocate more memory to the JVM to handle the required load through the `-Xms` and `-Xmx` Java switches, as follows:

```
java -Xms<memory size>m -Xmx<memory size>m -jar oc4j.jar
```

These switches explicitly set the amount of memory allocated for the initial and maximum Java heap size. The amount specified should be based on the available resources. The initial value should be equal to the maximum value to ensure the best performance. You should set the values to at least 256 MB.

For example, the following sets the initial and maximum Java heap sizes to 256 MB when starting the Mobile Server:

```
java -Xms256m -Xmx256m -jar oc4j.jar
```

3.4 File Deployment for Offline Instantiation

Chapter 16 discusses how to use Offline Instantiation. Section 16.8 discusses the deployment process. In this release, you cannot use file deployment for Offline Instantiation.

3.5 Modifications for Configuring and Using SSL

Chapter 17 discusses how to configure and use SSL communication. In order to use SSL properly in Oracle Database Lite, be aware of the following issues:

3.5.1 Which XML File Do You Modify for SSL?

In Section 17.2.2, "Configuring SSL for Standalone Mobile Server," it states to copy the `default-web-site.xml` file to `secure-web-site.xml`, before making your modifications. In this release, there is no `default-web-site.xml` file. Instead, copy the `http-web-site.xml` file and with the name of `secure-web-site.xml` file.

3.5.2 Using an SSL Temporary Certificate with mSync

mSync requires a Valid Certificate on an SSL Server with which to Sync. The Sample Certificate provided with the Default Install does not support Synchronizations. You must either provide a valid certificate or do not use SSL.

3.5.3 What is the Password for the SSL Sample Keystore?

Oracle Database Lite includes a sample keystore for SSL communication. The password for the SSL sample keystore is `oracle`.

3.5.4 Setting SSL for Java Application Communication

In Section 5.5 of the *Oracle Database Lite Developer's Guide for Java*, it describes how to enable SSL for synchronization of a Java client application. You set the `syncParam` in the `SyncOption` class to have the name/value pair of `security=SSL`. Note that the TransportType is always HTTP; there is no HTTPS setting. When you set `security=SSL`, the transport will use HTTPS even though it is set to HTTP.

3.6 Using Advanced Encryption Standard (AES) to Enable Security for mSync

In Section 5.5 of the *Oracle Database Lite Developer's Guide for Java*, it describes how to enable security for synchronization of a Java client application. Set the `syncParam` in the `SyncOption` class to designate the type of security for synchronization. To use AES for security, set the name/value pair of `security=AES`.

3.7 Enabling Device Software Updates

This section describes how to enable software updates and patches for your devices. It should be located in Chapter 10 in the *Oracle Database Lite Administration and Deployment Guide*.

- Section 3.7.3, "Controlling Device Software Updates"

3.7.1 Enabling Major Software Updates for Your Device

In order to facilitate a major software update, the corresponding `INF` file must be modified to reflect the new version number. Mobile Server relies on the application version number to determine if the client software is out-of-sync.

To update your software, perform the following:

1. In the software `INF` file, the administrator modifies the version number for the application to the current version, as follows.

   ```
   <setup name="Application Name" version="1.2.3">
   ```

2. The client user synchronizes with Mobile Server. Alternatively, the client user can invoke `update.exe` to check for the latest version of the software.

When the client user synchronizes, Mobile Server compares the client application software version number against the version number in the `INF` file. If the version numbers are different, Mobile Server compares the 'Last Modified Time' of all of the client application files against the server application files to determine the changes and then sends the modified files to the client device.

3.7.2 Applying Patches or Minor Updates

In order to apply specific patches to an existing installation for your application, the application developer creates an `INF` file with the `patch` attribute and copies it to the correct platform patch directory, each of which is located in the *`ORACLE_HOME`*`\j2ee\home\applications\mobileserver\setup\dmc` directory in the Mobile Server.

The following application `INF` file defines the `patch` element as `myFirstApp` for applying a patch to "Application Name" software:

```
<setup name="Application Name" version="1.2.3">
     <property>
        ...
        <patch>myFirstApp</patch>
    </property>
    ...
</setup>
```

The value in the patch element is a user-defined name. This will be the name of the directory into which the updates for the application are copied. In this example, the updates for "Application Name" are copied into the `myFirstApp` directory.

> **Note:** Be careful to have a unique patch directory name for each application. If you have the same directory name in the `patch` element, then all applications with that patch directory name receive the updates placed there.

In order to update a patch with the `olobj40.dll`, update the patch `INF` file as follows:

```
<setup name="Application Name" version="1.2.3" id='1001'>
   <install>
     <action msg_i='$FILE_I$' msg_u='$FILE_U$'>file</action>
```

```
      <file>
        <item>
          <src>/common/win32/olobj40.dll</src>
          <des>$APP_DIR$\bin\olobj40.dll</des>
        </item>
      </file>
   </install>
</setup>
```

Note: For a full description of INF files and the elements within them, see the "10.3.2 Installation Configuration (INF) File" section in the *Oracle Database Lite Administration and Deployment Guide*.

There are two mandatory attributes in a patch `INF` file, as follows:

- The `INF` file contains a version number, which is the same as the application version number. In the above example, the version number (10.0.0.0.0) tells DMS that the patch is meant for the application version 10.0.0.0.0.
- The `INF` file must have an ID, which is used for determining patch dependencies. This ID can be any number you choose.

If you do have dependencies among patches, then you use the `dependency` element to indicate these dependencies. For example, the previous patch for `olobj40.dll` was configured with the ID of 1001. The following `INF` file configures another patch with ID of 1002. This patch defines a dependency on the patch for `olobj40.dll` by configuring the ID number of 1001 in the `dependency` element.

```
<setup name="Oracle Lite WIN32" version="10.0.0.0.0" id='1002'>
  <property>
    <dependency>1001</dependency>
  </property>
</setup>
```

Update the patch, as follows:

1. The administrator copies the patch `INF` files to the patch directory.
2. The administrator copies the new application files to the application directory.
3. The client user synchronizes with Mobile Server. Alternatively, the client user can invoke `update.exe` to check for the latest version of the software.

The Mobile Server checks for patches and sends all new patches to the client device.

3.7.3 Controlling Device Software Updates

The administrator can control what software updates occur either through modifying the `Upgradeable` switch in the Mobile Manager or by programmatically defining what types of updates can occur. There are two types of software update that you can control, as detailed in the following sections:

- Section 3.7.3.1, "Enabling Software Updates for the Oracle Lite Platform"
- Section 3.7.3.2, "Updating Application Software On Each Client"

3.7.3.1 Enabling Software Updates for the Oracle Lite Platform

You can disable the Oracle Database Lite platform updates for the devices through one of two methods:

- Setting `Upgradeable` to `No`—Change the value of `Upgradeable` in the Mobile Manager GUI to `No`, as follows:
  1. Select the Mobile Devices tab in Mobile Manager.
  2. Select the Platform tab to display all Oracle Lite Platforms.
  3. Click **Oracle Lite WIN32** platform to display its properties.
  4. Change the value of `Upgradeable` to `No`. Click **OK**.
- Setting the `UPDATE_SOFTWARE` attribute in the Resource Manager to `false`. This has to be set programmatically on the Resource Manager object, as follows:

```
rs.setAttribute (ResourceConst.UPDATE_SOFTWARE, "false");
```

If you want to disable any application updates, but continue to allow Oracle Database Lite platform updates, then set the `UPDATE_SOFTWARE_APPS` attribute to `false`.

For example, to set the `UPDATE_SOFTWARE_APPS` attribute to false, do the following:

```
rs.setAttribute (ResourceConst.UPDATE_SOFTWARE_APPS, "false");
```

For a full example of how to set the Resource Manager attributes, see *ORACLE_HOME*`\Mobile\Server\samples\devmgr\java\AppUpdate.java`.

3.7.3.2 Updating Application Software On Each Client

You can control whether a new version of an application software is downloaded on each client. Set the `UPDATE_SOFTWARE_APPS` policy attribute of the `User` object to one of the following values to specify what type of update that the client can receive:

- `Major`—The devices attached to this user receives only major software updates. This is the default.
- `Minor`—The devices attached to this user receives only minor software updates.
- `False`—The devices attached to this user does not receive any software updates.

```
user.setPolicy (ResourceConst.UPDATE_SOFTWARE_APPS, "Minor");
```

For a full example of how to set the Resource Manager attributes, see *ORACLE_HOME*`\Mobile\Server\samples\devmgr\java\UserPolicy.java`.

Example 1 Upgrading Devices Attached to a Specific User

Each user owns one or more devices. You can configure it so certain users do not receive the latest update. The default is that all devices attached to a user receive current updates.

For example, you have two users: John and Tom. You want John's devices to stay at the current version, which is Oracle Database Lite Win32 version 10.0.0.0.0; however, you want Tom's devices to upgrade to the new version, which is Oracle Database Lite Win32 version 10.1.0.0.0. Configure each user's devices, as follows:

- For John, configure the `update.software.apps` attribute to `Minor`.
- For Tom, configure the `update.software.apps` attribute to `Major`.

3.8 What Options Can I Use Together With mSync?

You can configure what priority to synchronize with. Currently, you cannot choose both `High priority` and `Force refresh` simultaneously, as mSync does not support synchronizing with both of these options together.

3.9 How Do I View Synchronization Logs?

Section 10.4.7.5 in the *Oracle Database Lite Administration and Deployment Guide* describes how to view synchronization logs with the Mobile Manager. Instead, you must send the `Retrieve synchronization log` command to the appropriate client.

3.10 Creating Custom Mobile Device Platforms

As discussed in Section 10.4.3.2, "Creating Custom Platform," you can create your own Mobile device platform through the Mobile Manager. The custom Mobile device platform are an extension to the existing platform.

Oracle Database Lite comes with a number of Oracle Lite Mobile device platforms, such as Oracle Lite WEB and Oracle Lite WIN32. You may extend an existing Oracle Lite Mobile device platform to suite your requirements.

In order for you to extend an existing Mobile device platform, you must create the custom Mobile device platform and then extend it, either programmatically or declaratively.

For each new custom Mobile device platform, you will perform the following actions. An example of how to implement these steps is discussed in Section 3.10.1, "Example of Creating a Custom Mobile Device Platform".

1. Modify the `webtogo.ora` file to include the directory where the custom platform binaries are located. This enables the directory where the custom binaries are located to be mounted.
2. Copy the custom platform binaries to this directory.
3. Create a platform-specific `INF` file for this new custom platform.
4. Extend the existing platform either declaratively or programmatically.
5. Install the extended platform.

3.10.1 Example of Creating a Custom Mobile Device Platform

This section describes how to extend an existing platform with an example. In this example, we extend the Oracle Lite WEB platform to include our own version of UIX binaries (`uix2.jar`) file. All binaries are maintained in directories separate from the directories where the Mobile Server repository exists.

1. Modify the *`ORACLE_HOME`*`\mobile\server\bin\webtogo.ora` file to include the directory where the custom platform binaries are located.

In order to mount our directory on the repository file system, modify the `webtogo.ora` file and add our directory as an alternate directory under the file system section. After modifying this file, re-start Mobile Server.

The following adds `ALTERNATE_DIR` with the directory where the custom platform binaries are located. In our example, the `uix2.jar` file is located in `d:\my_dir`.

```
[FILESYSTEM]
TYPE=OS
ROOT_DIR=D:\orant\ora90\j2ee\home\applications\mobileserver
ALTERNATE_DIR=d:\my_dir=/my_app
```

2. Copy the custom platform binaries to this directory. For our example, copy the `uix2.jar` binaries to `d:\my_dir`.
3. Create a platform-specific `INF` file for this new custom platform. Copy the sample `INF` file that comes with Mobile Server, which is located in the *ORACLE_HOME*`\mobile\server\samples\devmgr\inf` directory. In our example, copy the `my_platform.inf` file to the `d:\my_dir`. Modify this `INF` file, as follows:
   a. Modify the `<patch>` element to use the base platform patch (minor updates) information.
   b. Add the platform binary JAR file, which in this example is `uix2.jar`, in the `<file>` section.
   c. Modify the `<execute>` section. Disable the parent platform `<execute>` section and replace it with the new custom platform `<execute>` section.
   d. Remove the `<finish>` section, so the `setup.exe` exits immediately after installing our platform.

   For our example, the modified `INF` file is shown below:

```
<setup name='My Platform' version='10.0.0.0.0'>
    <property>
        <storage>40</storage>
        <memory>30</memory>
        <location/>
        <port>80</port>
        <prompt>
          <item file='olobj40.dll'>$OLITE_CLOSE$</item>
        </prompt>
        <patch base='10.0.0.0.0'/>
    </property>
    <install>
        <action msg_i="$FILE_I$" msg_u='$FILE_I$'>file</action>
        <action base='disable' msg_i="$EXECUTE_I$"
            msg_u='$EXECUTE_I$'>execute</action>
        <file>
            <item>
                <src>/my_app/uix/uix2.jar</src>
                <des>$APP_DIR$\bin\uix2.jar</des>
            </item>
        </file>
        <execute>
            <item>
                <file>$APP_DIR$\bin\webtogo.exe</file>
                <args>-h</args>
```

```
                <wait>WebToGoSetupExit/WebToGoSetupStop</wait>
            </item>
        </execute>
    </install>
</setup>
```

4. Extend the existing Oracle Lite WEB platform either declaratively or programmatically.

   To extend the platform programmatically, see the `ExtendPlatform.java` sample program for an example. However, we are going to extend the platform declaratively using Mobile Manager, as follows:

   a. Logon to Oracle Mobile Server.

   b. Select the Mobile Manager and start Mobile Manager.

   c. Select the Mobile Devices tab.

   d. Select the Platforms sub-tab.

   e. Select the radio button next to Oracle Lite WEB.

   f. Click **Extend**.

   g. Type in the platform name—such as My Platform;US. You must specify the language abbreviation as part of platform name.

   h. Change the Setup INF file—`/my_app/my_platform.inf`.

   i. Click **OK**.

5. Install the extended platform.

   Open the URL `http://<mobile_server>/webtogo/setup` and download `setup.exe` for the new platform. Execute the `setup.exe` for this platform.

4 Modifications to the Developer's Guide

The following details a modification that should be in the *Oracle Database Lite Developer's Guide*.

4.1 Troubleshooting Consolidator Sequences That Are Not Working

Section 3.5.6 describes the Consolidator Sequences. If the Consolidator Sequences do not work properly, check your parent publications. All parent publications must have at least one publication item. If you do not have any publication items for the parent publication, then create a dummy publication item within the parent.

5 Modifications to the Developer's Guide for Java

The following subjects detail modifications that should be in the *Oracle Database Lite Developer's Guide for Java*.

- Section 5.3, "Setting SSL for Java Application Communication"
- Section 5.4, "Truncating Tables and Synchronization"

5.1 Enable Registry Support for Web Applications

The registry entry is deprecated for the Web applications in this release. To enable the registry support for Web applications, add the following parameter in the `[WEBTOGO]` section of `webtogo.ora` file of Mobile Server and Mobile Development Kit installation. This enables Registry tabs in both the Packaging Wizard and Mobile Manager.

```
REGISTRY_TAB  = YES
```

5.2 Publishing PALM applications using Mobile Server Scripting Language

To publish a PALM application using Mobile Server scripting language (in the `.ini` file), the user performs the following:

1. Create a Palm directory in the application root directory.
2. Copy the `.prc` file to the Palm directory.

This is only required for Palm applications published using the Mobile Server scripting language.

For example, the user copies the `.prc` file to the `Palm_FormOrders/palm` directory:

```
[DIRECTORY]
Palm_FormOrders
Palm_FormOrders/palm
# Copy files
[COPY]
$PC$/Obj/FormOrders.prc    $IFS$/Palm_FormOrders/palm/FormOrders.prc
[APPLICATION]
NAME=Palm_FormOrders
DIRECTORY= /Palm_FormOrders
VIRTUALPATH=/Palm_FormOrders
DEFAULTPAGE=
CLASSPATH=
ICON=
DESCRIPTION=Sample Order Demonstration
…..
```

5.3 Setting SSL for Java Application Communication

In Section 5.5 of the *Oracle Database Lite Developer's Guide for Java*, it describes how to enable SSL for synchronization of a Java client application. See Section 3.5.4, "Setting SSL for Java Application Communication" for more information on how to enable SSL for a Java client application.

5.4 Truncating Tables and Synchronization

Records removed from the server through a `truncate` command will not be removed from the client unless a complete refresh is triggered. The truncate

command is considered a DDL operation. Consequently, the necessary DML triggers do not fire and therefore the operations are not logged for fast refresh.

6 Modifications to the Tools and Utilities Guide

The following subject should be in the *Oracle Database Lite Tools and Utilities Guide*.

6.1 String Overlap in Packaging Wizard UI for Windows 2003 Server

If the Packaging Wizard is running on a Windows 2003 machine, then, in some cases, the label on the platform screen may overlap with the dropdown control. This is a JDK bug specific to the Windows 2003 Platform. Increase the size of the Packaging Wizard window to fix the overlap issue.

6.2 Changing the Sequence Increment Value in the Packaging Wizard is Not Allowed Once the Application Is Published

If you edit an existing application in the packaging wizard, which uses sequences, then you can not modify the increment value of the existing sequence.

7 Modifications to the SQL Reference

The following sections discusses the limitations to SQL in Oracle Database Lite.

7.1 SQL Limitations for Oracle Database Lite

There are limitations to SQL that is different than the Oracle database. These are as follows:

Table 1 Datatype Limits

Datatypes	Limit	Comments
BFILE	Maximum size: 2 GB Maximum size of the directory or file names: no database imposed limit	All BFILE objects are stored as LOB
BLOB	Maximum size: 2 GB	
CHAR	Maximum size: 4096 bytes	
CHAR VARYING	Maximum size: 4096 bytes	
CLOB	Maximum size: 2 GB	
Literals	No limit	

Table 1 (Cont.) Datatype Limits

Datatypes	Limit	Comments
LONG	Maximum size: 2 GB	A table can have any number of long columns
NUMBER	Operating system limit	NUMBER is converted to a double precision number on the native platform
NUMBER (p, s)	999 ... (38 9's) x 10 ^ 125 maximum -999... (38 9's) x 10 ^125 minimum	Maximum precision of 38 decimal digits
VARCHAR	Maximum size: 4096 bytes	
VARCHAR2	Maximum size: 4096 bytes	

Table 2 Physical Database Limits

Item	Limit	Comments
Database Block Size	4096 bytes	Fixed size
Database File	1 database file for each catalog	An application can open any number of catalogs.
Database File Size	4 GB	Affected by the operating system. Maximum file size allowed by the operating system.
Max Object or Row Length	4040	When an object (row) exceeds this length, it is converted into a binary long object. So, UNION will not work on this table.
DSN Name	31 bytes (31 US chars)	Limit is 31 bytes
Database Path Name	129 bytes	_MAX_PATH -5, which is 255 on Win32
Database filename	129 bytes	_MAX_PATH -8, which is 252 on Win32.

Table 3 Logical Database Limits

Item	Limit	Comments
Indexes	Maximum for each table	unlimited
Columns	table	1000
	index	32 columns maximum
Constraints	Maximum for each column	unlimited
Nested queries	Maximum number	unlimited
Rows	Maximum number for the table	no limit
SQL statement length	Maximum length of statements	unlimited, particular tools may impose lower limits

Table 4 Process/Runtime Limits

Item	Limit	Comments
Shared Memory	128 MB Maximum	
Cache	64 4K blocks by default	Used for caching database pages

7.2 Embedded SQL Commands

Embedded SQL is no longer supported by Oracle Database Lite; thus, table 1-4 and the text preceding the table in Section 1.2.4, "Commands" is not valid.

8 Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

http://www.oracle.com/accessibility/

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.