

Understanding The Oracle Lite Database

*An Oracle White Paper
June 2007*

Understanding The Oracle Lite Database

Oracle Lite Database Overview	3
Using the Oracle Lite Database In Embedded Applications.....	3
Small Embedded Database for Application Solutions	3
Multiple Processes Sharing the Same Oracle Lite Database	4
Multi-User Service for Oracle Database Lite.....	5
Embedding the Oracle Lite Database in Your Application	5
Storing Data in Oracle Database Lite.....	6
Oracle Database Lite Features.....	6
Oracle SQL Support.....	6
Running SQL-92 on Oracle Database Lite	6
Oracle Database Lite Datatypes	7
Character Data	7
Numeric Data	7
Binary Data.....	8
Date and Time	8
Concurrency Support With Transactions in Oracle Database Lite	8
Nested Transaction Support	9
Database Constraint Support in Oracle Database Lite	9
Other Important Features	9
Accessing Data in the Oracle Lite Database	10
Using JDBC to Access Data	10
ODBC	10
ADO.Net	10
Security in Oracle Database Lite	10
Encrypting the Oracle Lite Database.....	11
Defining Which Users Can Access the Data	11
Accessing Database Objects.....	11
Conclusion.....	11

Understanding The Oracle Lite Database

ORACLE LITE DATABASE OVERVIEW

The Oracle Database Lite RDBMS is a small footprint, Java-enabled, secure, relational database management system created specifically for mobile phones, laptop computers, and handheld computers. The Oracle Database Lite RDBMS can be used on Windows 2003/XP/Vista, Windows CE/Windows Mobile, Linux, and Symbian platforms. The Oracle Lite database is simple to embed into your application to provide seamless access to your data. Also, because it is a small database specifically designed for a client device, it is easy to administer. You can build applications from a variety of programming languages, such as Java, C, C++, C#, and Visual Basic and access the database with the JDBC, ODBC, or ADO.Net interfaces. If you are used to working with stored procedures, we have enabled stored procedures for Java, C++ and C#.

The Oracle Lite database is compliant to the SQL92 standard and compatible with Oracle databases. In addition, it is compliant to the ACID requirements for transaction support. If you do interact with an Oracle database, Oracle Database Lite is upwards compatible; that is, queries that execute against Oracle Database Lite will also execute against the Oracle database.

You can use the Oracle Lite database either with the Mobile client—which uses synchronization to replicate data between the client and the Oracle database—or you can embed the Oracle Lite database within an independent application of your own design. Either way, the application can use the Oracle Lite database to contain and manage the client data.

USING THE ORACLE LITE DATABASE IN EMBEDDED APPLICATIONS

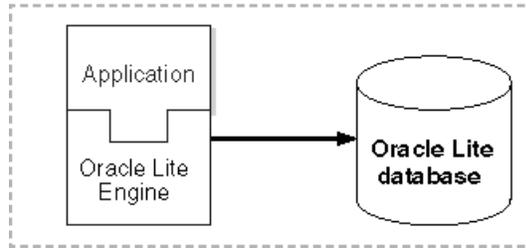
The following sections describe the several options on how to use the Oracle Lite database for your application needs.

Small Embedded Database for Application Solutions

Often, standalone applications need to locally store or manipulate a large amount of data. You can decrease the amount of development effort needed for your application by embedding an existing database in your application implementation. Oracle Database Lite contains a proven implementation of a small, reliable, relational database that provides many of the functions necessary to manage your data in a small application or on hardware. The following picture shows an

application integrated with the Oracle Lite engine, and the Oracle Lite database in a single process:

Embedded Application Process



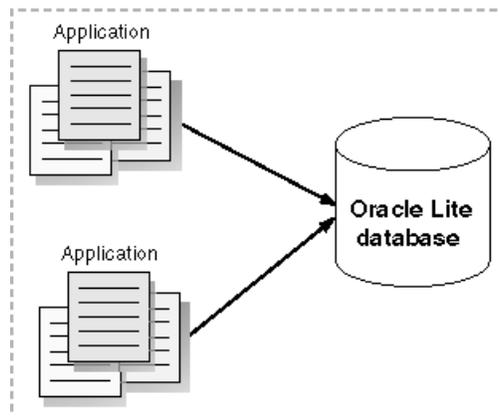
The following are examples of how you might use the Oracle Lite database in your application:

- An application may need an independent small, relational database to exist solely for the application's need to store, search and retrieve application data. For example, if you have an individual accounting application or an address book, it may need a small embedded database to store the data.
- The software that supports hardware may need an embedded database to facilitate gathering information, which can be retrieved by a service technician when the unit is serviced. Examples of this type of implementation may include any vending machine that tracks current inventory or software integrated into a car that monitors statistics on the automobile's performance.

Multiple Processes Sharing the Same Oracle Lite Database

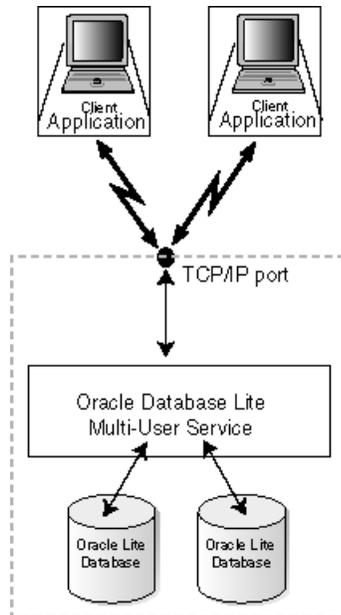
You can configure multiple application processes to share the same database on the same machine. Thus, when each application is launched, each application exists in its own process and can access the same database independently. In this scenario, Oracle Database Lite libraries use shared memory to coordinate locking between both processes.

Multiple Processes Accessing Database



Multi-User Service for Oracle Database Lite

There may be a time when you want to protect all of your data on a centralized machine and only allowing the clients to access the information remotely. You can centralize your Oracle Lite databases by installing them on a Windows or Linux host machine. The Oracle Database Lite Multi-User Service enables remote applications to connect to the Oracle Lite database over a TCP/IP connection. The Multi-User Service handles connections to multiple databases; the client application uses the provided DSN to control to which database it wants to connect. The following picture shows the Oracle Lite databases installed on a single host machine where one or more clients may access the data.



Embedding the Oracle Lite Database in Your Application

You can easily embed the Oracle Lite database with your application by copying five files from the Mobile Development Kit and packaging them with your application.

For example, on the Windows platform, copy the following files from the Mobile Development Kit library, which is located in `ORACLE_HOME/Mobile/Sdk` directory, into same directory as your application DLLs, as well as in your `PATH`.

- `olite40.msb`: Oracle Database Lite message file
- `olobj40.dll`: Oracle Database Lite object kernel
- `olod2040.dll`: Oracle Database Lite ODBC driver
- `olsql40.dll`: Oracle Database Lite SQL runtime library
- `olstdll.dll`: Oracle Lite Common library

These five files allow you to access the database using the ODBC interface. If you are using the Multi-User Service, using any Java program with the database, or if you want to use any of the database utilities, you may need to copy over other files.

STORING DATA IN ORACLE DATABASE LITE

The Oracle Lite database stores and retrieves the user data specific to the device. It manages your application data in the following two files:

- ODB file: Most of the data is stored in the Oracle Lite file with an ODB extension.
- OBS file: Any BLOB objects—either binary or character—and the indexes are stored in a file with an OBS extension.

ODB files are limited to 4 GB. The maximum size for the OBS file is 16 terabytes—or the operating system limit for files—whichever is less. Because BLOB data and indexes are stored in an OBS file, this provides virtually unlimited storage space for BLOB data.

ORACLE DATABASE LITE FEATURES

Oracle Database Lite is a small, relational database with concurrency support that you can use as an embeddable database for your application. The following sections detail some of the advanced features that Oracle Database Lite supports, which you may not find in other small embedded databases.

Oracle SQL Support

Oracle Database Lite uses Oracle SQL as its default SQL language. When you use Oracle SQL with Oracle Database Lite, then several SQL-92 features are supported including the following:

- Column datatypes: TIME, TIMESTAMP, TINYINT, and BIT
- Expressions: CASE and CAST

Running SQL-92 on Oracle Database Lite

If you want to support SQL-92 instead of Oracle SQL, then you can change the SQL compatibility by adding the following parameter in the POLITE.INI file:

```
SQLCOMPATIBILITY=SQL92
```

Oracle Database Lite Datatypes

The following sections detail the datatypes that Oracle Database Lite supports for character, numeric, binary, and date/time.

Character Data

For characters, you can use the following:

- CHAR, VARCHAR or VARCHAR2—You may define the length where the minimum is 1 byte and a maximum is 4,096 bytes. You must specify a length for VARCHAR2.
- Long Character data. Store up to 2 GB of character data in columns using the LONG VARCHAR or CLOB datatypes.

Numeric Data

For numbers, you can have either integer or decimal type numbers.

- Integers—For integers, you can use INTEGER, BIGINT, SMALLINT, and TINYINT datatypes. You define the precision of the INTEGER and SMALLINT. The BIGINT datatype has a precision of 19 digits; TINYINT has a precision of 1 byte.
- Decimal numbers—For decimal numbers, you can use DECIMAL, NUMBER, NUMERIC, FLOAT, REAL, OR DOUBLE PRECISION datatypes. You can define the precision (decimal value or the total number of digits in the number including numbers on both sides of the decimal point) or scale (fractional value or number of digits after the decimal point) for the the DECIMAL, NUMBER or NUMERIC datatypes.

When you define the DECIMAL, NUMBER, or NUMERIC datatype, you can define them in one of three ways:

- Without precision or scale: If you use the datatype without defining any precision or scale, then, each of these datatypes default to a precision of 38 and a scale of 0. For example, NUMBER would define a number up to 38 digits and no digits after the decimal point.
- With precision definition: If you define the precision only, the scale defaults to 0. For example, NUMBER(10) would define an integer up to 10 digits and no digits after the decimal point.
- With both precision and scale definition: You can define both the precision and scale for the datatype. For example, if you define NUMBER (5, 2), then the number can be a total of 5 digits, which concludes to 3 digits before and 2 digits following the decimal point. The following are the legal and illegal values that can be contained in this variable:

- Illegal values: 12345, 1234
- Legal values: 1, 123, 123.12, 123.123, which will be rounded to 123.12

Oracle Database Lite defines the precision for FLOAT, REAL, or DOUBLE PRECISION depending on each platform.

Binary Data

For binary data, you can use the BINARY, VARBINARY, BIT and RAW datatypes. Store a single bit unconstrained by character semantics in the BIT datatype. Store binary data up to 4,096 bytes in either the BINARY or RAW datatype; however, you must specify the size for the RAW datatype. And with the VARBINARY datatype, you can store variable binary data that is not interpreted.

For long binary data, you can store up to 2 GB of binary data in columns in the LONG, LONG VARBINARY, LONGRAW, or BLOB datatypes.

Date and Time

For date and time datatypes, you can use the DATE, TIME, and TIMESTAMP datatypes. The TIME datatype stores a time value in hours, minutes, and seconds. The TIMESTAMP datatype stores the year, month, and day values of a date and the hour, minute, second value of time.

Concurrency Support With Transactions in Oracle Database Lite

Oracle Database Lite supports concurrent transactions. Each transaction is isolated from another. Even though many transactions run concurrently, transaction updates are concealed from other transactions until the transaction commits. You can specify what level of isolation is used within the transaction, as listed in the following table:

Isolation Level	Description
Read Committed	A READ COMMITTED transaction first acquires a temporary database level read lock, places the result of the query into a temporary table, and then releases the database lock. During this time, no other transaction can perform a commit operation. No data objects are locked. All other transactions are free to perform any DML operation—except commit—during this time. A READ COMMITTED transaction provides the highest level of concurrency, as it does not acquire any data locks and does not block any other transaction from performing any DML operations.
Repeatable Read	A query acquires read locks on all of the returned rows, which are held until the end of the transaction. More rows may be read locked because of the complexity of the query itself, the indexes

	defined on its tables, or the execution plan chosen by the query optimizer.
Serializable	Acquires shared locks on all tables participating in the query. The same set of rows is returned for the repeated execution of the query in the same transaction. Any other transaction attempting to update any rows in the tables in the query is blocked.
SingleUser	Only one connection is permitted to the database. The transaction has no locks and consumes less memory.

Nested Transaction Support

Oracle Database Lite supports nested transactions. A nested transaction is a method where you can define one or more markers, known as savepoints, at certain points within the transaction. You can define as many savepoints in a transaction as you want.

Savepoints enable you to provide effective error recovery for your transaction by specifying what parts of the transaction can be saved and avoiding the potential of losing the entire transaction. If an error occurs within the transaction, you can revert the transaction back to the last savepoint.

Database Constraint Support in Oracle Database Lite

Oracle Database Lite supports the numeric, character or date-based primary key. Primary keys may consist of a single or multiple columns. You can use the foreign key constraint to enforce referential integrity between tables. If you specify the `ON DELETE CASCADE` clause when creating the foreign key constraint, then detail records are automatically deleted if the corresponding master record is deleted. Otherwise, you must first delete the master record before you can delete the detail records.

Oracle Database Lite can help you ensure that all values in a particular column or combination of columns are unique. Once you create a unique constraint for the particular column or columns, Oracle Database Lite automatically prevents the creation of duplicate values.

If you do not want any column to contain null values, then you can specify a `NOT NULL` constraint for that particular column. Insert and update operations on columns that have `NOT NULL` constraints automatically fail if you attempt to either insert or update a null value for that particular column.

Other Important Features

- Stored procedures and trigger support in Java, C++ or C#.
- Alter table support. You can make changes to an existing table. For example, you can add/modify a column or add/delete table constraints.

- Writing to views. If you create a view based on a single table, then any insert, update and delete operations on the view are automatically applied to the underlying table—provided that the operation does not violate any table constraints.
- You can create SQL statements that use CASE or CAST expressions

ACCESSING DATA IN THE ORACLE LITE DATABASE

You can use JDBC, ODBC, or ADO.Net to access data in the Oracle Lite database.

Using JDBC to Access Data

The Java Database Connectivity (JDBC) interface specifies a set of Java classes that provide an ODBC-like interface to SQL databases for Java applications. JDBC, part of the JDK core, provides an object interface to relational databases. Oracle Database Lite conforms to the JDBC 1.2 API specification standard.

Oracle Database Lite supports JDBC through an Oracle Database Lite Type 2 and Type 4 JDBC drivers that interpret the JDBC calls and pass them to Oracle Database Lite . The Type 4 JDBC driver can only be used with the multi-user service.

For Mobile clients, all JDBC drivers are provided for you to use within the Oracle Database Lite binaries. However, for embedded applications, you must include the correct binaries when you package the application.

ODBC

The Microsoft Open Database Connectivity (ODBC) interface is a procedural, call-level interface for accessing any SQL database, and is supported by most database vendors. It specifies a set of functions that allow applications to connect to the database, prepare and execute SQL statements at runtime, and retrieve query results.

Oracle Database Lite supports Level 3 compliant ODBC 2.0 and the ODBC 3.5 drivers through Oracle Database Lite ODBC drivers.

ADO.Net

Oracle Database Lite supports ADO.NET for Microsoft .NET and Microsoft .NET Compact Framework. The Oracle Database Lite ADO.NET provider resides in the `Oracle.DataAccess.Lite` namespace.

SECURITY IN ORACLE DATABASE LITE

The Oracle Lite database provides numerous security features to ensure that you have a secure place to store your data and a secure method to limit access to that data.

Encrypting the Oracle Lite Database

Once the data is in the database, you may want to ensure that no one can read what is in the database. To secure the data at the file level, you can encrypt the Oracle Lite database. Once encrypted, the data stored in the database files cannot be interpreted by examining the files. When a validated user accesses the database, the data is automatically decrypted by Oracle Database Lite in memory.

Oracle Database Lite uses the Advanced Encryption Standard (AES) encryption. If you do not want to use AES encryption, then you can insert your own encryption module.

Defining Which Users Can Access the Data

You can create multiple user accounts in the database, so that each user has his or her own database account. Each database user account must have a password. You can create multiple users in your Oracle Lite database for your embedded application with the CREATE USER command.

To make granting access to data easier for the administrator, any user can be assigned one or more database roles. A role controls the type of operations a user can do, such as creating other database users, new database objects and assigning privileges to other users. Oracle Database Lite has three pre-defined roles: ADMIN, RESOURCE and DBA.

When you create a database using the CREATEDB utility or the CREATE DATABASE command, Oracle Database Lite creates a special user called SYSTEM, which has all database privileges.

Accessing Database Objects

By default, users can access only the database objects that they own. A user can explicitly grant other users access to his/her database objects. Access is granted for each database object—as well as for each operation type. This enables the grantor to define specifically who can do what. For example, the grantor can assign select, insert, update or delete privileges on a table object separately. Thus, the grantor can control which users can perform select and update commands and so on.

CONCLUSION

The Oracle Database Lite features provide a small, embedded, rich database that supports SQL92, stored procedures, and transactional concurrency. In addition, it is easy to embed into your existing applications and supports multiple deployment scenarios. Lastly, the security integrated into the database keeps it safe from unauthorized access.



Understanding The Oracle Lite Database
June 2007

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2007, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.