

Replication with Personal Oracle Lite™

Release 2.4

Part No. A53900-01

ORACLE®

Enabling the Information Age

Replication with Personal Oracle Lite, Release 2.4

Part No. A53900-01

Copyright © 1997, Oracle Corporation. All rights reserved. Printed in the U.S.A.

Authors: Anne Bers, Vince Morton

Contributors: Joe Flood, Jia-Der Day, Mark Gaydos, Paul Beesley, Denise Lahey

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, back up, redundancy and other measures to ensure the safe use of such applications if the programs are used for such purposes, and Oracle disclaims liability for any damages caused by such use of the Programs.

This software/documentation contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

Restricted Rights Legend

Programs delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication and disclosure of the Programs shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free.

Personal Oracle Lite is a trademark of Oracle Corporation, Redwood City, California. ORACLE, SQL*Net, and SQL*Plus are registered trademarks of Oracle Corporation, Redwood City, California.

All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.



Read This First!

This section provides important information on how to use *Replication with Personal Oracle Lite*. It describes:

- the intended audience for this guide
- how to use this guide
- a fast path through this guide
- notational conventions used in this guide
- where to find more information
- where to send your comments

Note: If you're an advanced user, see the "Fast Path" section for a shortcut through this guide.

Audience

Replication with Personal Oracle Lite is written for application developers who want to create applications that:

- are optimized for single-user databases on small-footprint devices
- utilize a three-tiered client/agent/server architecture, providing efficient access to corporate data from outside the enterprise
- allow periodic updates to local information from an enterprise database, or vice versa
- support wireless, dial-up, or local area network (LAN) connectivity
- enable mobile or distributed users to dispatch multiple transactions without waiting, or disconnect and continue working when network coverage is marginal or absent
- run under Windows 3.1, Windows 95, or Windows NT

This manual introduces replication concepts, describes Personal Oracle Lite's replication capabilities, and provides instructions on how to set up an environment for developing and deploying replication applications with Personal Oracle Lite.

How to Use This Guide

Here are some suggestions for how to get the most out of this guide given your level of experience with replication and Personal Oracle Lite:

If you are:	Follow this path:
New to replication and Personal Oracle Lite	<p>Read Chapter 1, “Understanding Replication,” for an overview of Personal Oracle Lite replication concepts.</p> <p>Work with your database administrator to prepare the master and snapshot sites for replication, as described in Chapter 2, “Preparing for Replication.”</p> <p>Read Chapter 3, “Using Replication,” for steps on how to perform replication operations programmatically or through the Navigator user interface.</p> <p>Review Appendix B, “Replication OLE Control Methods,” to familiarize yourself with the Replication OLE Control.</p>
Familiar with replication and Personal Oracle Lite, but new to developing replication applications with Personal Oracle Lite	<p>See the “Fast Path” section, below, for the minimum steps required to ensure successful replication with Personal Oracle Lite.</p>

Fast Path

This section provides advanced users with a shortcut through this guide. The following table lists the minimum steps required to ensure successful replication with Personal Oracle Lite, and where the instructions are found in this guide.

These steps:	Are described in:
<ol style="list-style-type: none">1. Prepare the master site for replication:<ul style="list-style-type: none">- verify Oracle7 server support for replication- set up support for the replication engine- create snapshot logs (if using fast refresh)- prepare master tables for replication	“Preparing the Master Site for Replication,” in Chapter 2.
<ol style="list-style-type: none">2. Prepare the snapshot site for replication in one of two ways:<ul style="list-style-type: none">- add integrity constraints to snapshot base tables after creating snapshots, or- pre-create snapshot base tables	“Preparing the Snapshot Site for Replication,” in Chapter 2.
<ol style="list-style-type: none">3. Set up replication connectivity:<ul style="list-style-type: none">- choose whether to use SQL*Net or Oracle Mobile Agents- install and configure the connectivity software	“Setting Up Replication Connectivity,” in Chapter 2.

Notational Conventions

The following notational conventions are used in this guide:

UPPERCASE

Uppercase characters represent command names, SQL reserved words and keywords, or filenames.

Bold

Bold text indicates items within a menu, buttons, or tabs on a property sheet.

<i>Italics</i>	Italic text is used for emphasis or to represent variables. For variables, substitute the appropriate values.
Monospace text	Monospace text represents commands or SQL statements. Type the text exactly as shown.
[]	Brackets enclose optional items. (Do not enter the brackets.)
{ }	Braces represent a set of options from which you must choose at least one. (Do not enter the braces.)
	A vertical bar represents an “or” option among several options. You enter only one of the options. (Do not enter the vertical bar.)

For More Information

The following Oracle document provides additional information about replication:
Oracle7 Server Distributed Systems, Volume II: Replicated Data, A32545–2.

To learn more about developing applications with Personal Oracle Lite, refer to the
Personal Oracle Lite Programmer’s Guide, A52614-03.

Your Comments Are Welcome

We value your comments as an Oracle user and reader of our manuals. As we write, revise, and evaluate our documentation, your opinions are the most important input we receive. At the back of this manual is a Reader’s Comment Form. We encourage you to use it to tell us what you like and dislike about this (or other) Oracle manuals.

If the form is missing, or if you would like to contact us, write us at:

Personal Oracle Lite Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065



Contents

Chapter 1	Understanding Replication	1
	Overview	2
	Read-Only vs. Updatable Snapshots	3
	Snapshot Transaction Logs	4
	Snapshot Refresh	5
	Refresh Types	5
	Refresh Group	6
	Error Conditions During Refresh	6
	Refresh-In-Progress Mode	7
	Conflict Detection and Resolution	7
	Replication OLE Control (OCX)	8
Chapter 2	Preparing for Replication	11
	Preparing the Master Site for Replication	12
	Verify Oracle7 Server Support for Replication	12
	Set Up Support for the Replication Engine	12
	Create Snapshot Logs (for Fast Refresh)	13
	Prepare Master Tables for Replication	13
	Preparing the Snapshot Site for Replication	13
	Pre-Creating Snapshot Base Tables	14
	Setting Up Replication Connectivity	15
	SQL*Net Replication	15
	Oracle Mobile Agents Replication	16
Chapter 3	Using Replication	19
	Overview	20

Connecting to Databases	20
Working with Snapshots	21
Creating Snapshots	21
Adding Snapshots to the Snapshot Refresh Group	22
Refreshing Snapshots	22
Dropping Snapshots	22
Scheduling Refreshes	23
Handling Errors During Refresh	23
Working with Oracle Mobile Agents	24
Handling Notification	24
Using the Long Transaction Feature	24
Deploying Replication Applications	25
Distribute the Required Software	25
Clearing Snapshot Transaction Logs	26
Recreate Snapshots	26
Running a Sample Application	27
Troubleshooting	27

Appendix A	Setting Up Oracle Mobile Agents	29
	Configuring the Oracle Mobile Agents Client Software	30
	Configuring the Oracle Mobile Agents Server Software (for DBAs)	30

Appendix B	Replication OLE Control Methods	33
	Opening a Connection to the Master Site	34
	Opening a Connection to the Snapshot Site	34
	Initializing Replication Service (required, for Oracle Mobile Agents)	34
	Creating a Snapshot	35
	Resetting the Refresh Group	36
	Adding Snapshots to the Refresh Group	36
	Refreshing the Refresh Group	36
	Dropping a Snapshot	37
	Retrieving Error Information	37
	Closing the Connection to the Master Site	37
	Closing the Connection to the Snapshot Site	38
	Enabling Long Transaction Mode (Advanced Feature)	38
	Getting the Long Transaction Status (Advanced Feature)	38
	Resuming the Replication Operation (Advanced Feature)	38
	Aborting the Replication Operation (Advanced Feature)	39

Appendix C	Sample Replication Application	41
	About this Application	42
	1. Connecting to the Master Site	43

1.1. Check whether OMA Connection is selected	43
1.2. Initialize replication service, if OMA Connection is selected	43
1.3. Connect to the master site	43
2. Connecting to the Snapshot Site	44
3. Performing Replication Operations	44
3.1. Initiate an instance of the Replication OLE control	45
3.2. Create a snapshot (calls Create Snapshot form, item 4.)	45
3.3. Reset the snapshot refresh group	45
3.4. Add a snapshot (calls Add Snapshot form, item 5.1)	45
3.5. Refresh the snapshot refresh group	45
3.6. Get error information	46
3.7. Drop a snapshot (calls Drop Snapshot form, item 5.2)	46
3.8. Handle notification (if using Oracle Mobile Agents)	46
3.9. Enable the long transaction feature	47
3.10. Get long transaction status	47
3.11. Resume the pending operation	47
3.12. Abort the pending operation	47
4. Creating a Snapshot	48
5. Adding/Dropping Snapshots	49
5.1. Add a snapshot to the refresh group	49
5.2. Drop a snapshot	49

1

Understanding Replication

This chapter introduces Personal Oracle Lite replication concepts, including:

- overview of replication
- read-only vs. updatable snapshots
- snapshot transaction logs
- snapshot refresh types, groups, and error conditions
- refresh-in-progress mode
- conflict detection and resolution
- Replication OLE control (OCX)

Note: For steps on how to perform the replication operations described in this chapter, see Chapter 3, “Using Replication.”

Overview

Personal Oracle Lite enables you to perform replication by creating local copies, or **replicas**, of data from a primary remote database. Local copies of data are called **snapshots** because the information is captured at a moment in time. Once a snapshot has been created, it can be periodically **refreshed** to reflect a more recent state of the data.

Throughout this document the primary remote database, or Oracle server, is referred to as the **master site**, and the tables at the master site are referred to as the **master tables**. The local database, or Personal Oracle Lite client, is referred to as the **snapshot site**.

Although it sounds fairly simple, replication relies on precise coordination between the master site and the snapshot site. For an Oracle server to function as a master site, it must be version 7.3 or later, and it must be configured by the database administrator to handle replication services. If the master site is not configured properly, replication cannot occur.

A Personal Oracle Lite database can function as a snapshot site, only. It supports two types of snapshots: read-only and updatable. A **read-only snapshot** can be used only for queries. Only changes to the master table are replicated down, or pulled, to the snapshot; no changes can be made to the snapshot. An **updatable snapshot** provides an updatable copy of a master table and can be defined to contain a full copy of a master table or a subset of rows in the master table that satisfy a value-based selection criteria. Changes can be made to the snapshot and propagated back, or pushed, to the master table.

A further distinction is made based on the complexity of the snapshot's defining query. A **simple snapshot** is based on a single remote table and has none of the following: distinct or aggregate functions; GROUP BY or CONNECT BY clauses; subqueries; joins; or set operations. If a snapshot's defining query contains any of these clauses or operations, or is derived from more than one master table, it is a **complex snapshot**. A read-only snapshot can be either simple or complex. Updatable snapshots can be only simple. To summarize:

Read-only snapshots:	Updatable snapshots:
are for queries only	are for queries and updates
can be simple or complex	must be simple

Read-Only vs. Updatable Snapshots

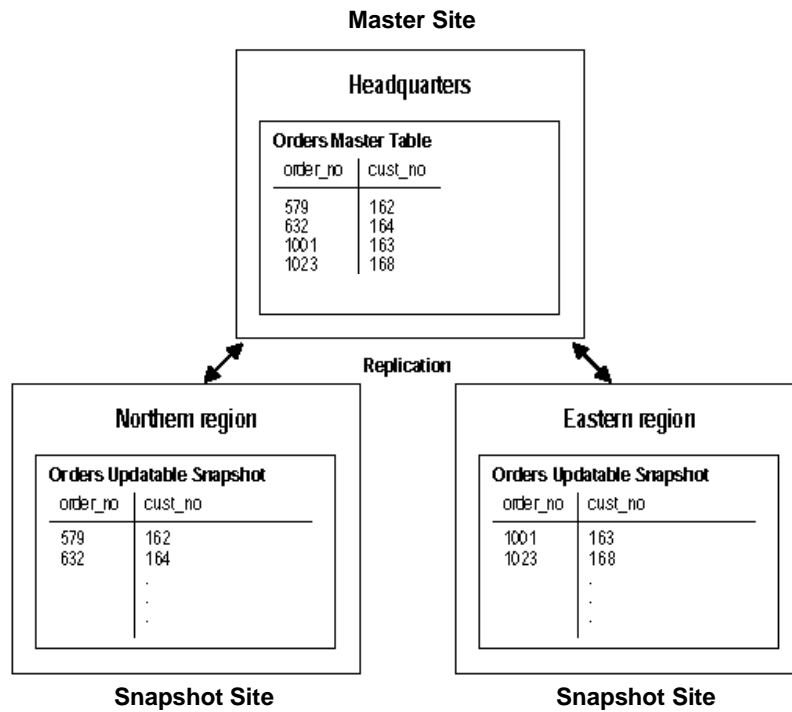
A read-only snapshot is a full copy or subset of rows of a master table that reflects a recent state of the master table. It is defined by a query that references one or more master tables, views, or other snapshots. Only changes to the master table are pulled to the snapshot; no changes can be made to the snapshot. This is also called uni-directional replication.

Benefits of working with a read-only snapshot rather than the master table itself include:

- Faster response time because the requested data does not have to travel across a network
- If the master site becomes unavailable (due to network failure, for example), you can continue to query the data.
- Allows mobile or distributed users to work disconnected from the master site.

Like a read-only snapshot, an updatable snapshot is either a full copy or subset of rows of a master table that reflects a recent state of the master table. However, the rows of an updatable snapshot must be derived from a single master table. In addition, during a refresh of an updatable snapshot, changes made to both the snapshot site and master site are propagated in both directions. This is called bi-directional replication.

A single master site can be used to consolidate information provided from multiple updatable snapshot sites. In the figure below, orders can be entered at each of the sales offices, but all orders are processed at corporate headquarters. You can think of the ORDERS snapshot at each of the sales offices as a writable subset of the of the ORDERS table at headquarters.



Snapshot Transaction Logs

At the snapshot site, a table called `USLOG` contains a summary of all updatable snapshot transactions at the snapshot site. In addition to `USLOG`, a separate log for every snapshot is stored in tables called `USLOG$_snapshotname`. These logs contain the full details of every transaction for the associated snapshot.

Note: These tables can be helpful for debugging replication applications, but they should not be manually altered. The exception is when preparing a starter database for deployment. For more information, see “Deploying Replication Applications, in Chapter 2.

Snapshot Refresh

You can refresh read-only snapshots whenever you want to reflect a more recent state of the master tables from which they are derived. Similarly, you can refresh updatable snapshots when you want to push changes from the snapshot site to the master site, and vice versa. During the snapshot refresh process, the following events occur in sequence:

1. Any changes to updatable snapshots are pushed to the master site.
2. The master site applies the valid transactions to the master tables, and the invalid ones are recorded in the error queue.
3. Snapshots are updated with the latest data, which is pulled from the master site.

To preserve transaction order, changes in snapshots pushed to the master site during refresh must occur in the same sequence as they were performed initially in the snapshot. Personal Oracle Lite handles this requirement automatically.

Restrictions on Refreshing Updatable Snapshots

If an updatable snapshot is based on a query that contains a restriction predicate in the WHERE clause, it will contain rows that reflect this restriction. However, it is still possible to insert or update rows in the snapshot that violate this restriction. For example, if an updatable snapshot is defined as follows:

```
select * from CUSTOMERS where zip_code = 94064
```

it is possible to insert or update rows in the snapshot in which `zip_code` is a value other than 94064. These updated rows will remain in the snapshot, and may be updated, deleted, and pushed to the master site. However, once these rows have been pushed to the master table and the snapshot refreshed, the rows that violate the query restriction will not be pulled back to the snapshot. You may want to design your application to handle this restriction.

Refresh Types

Personal Oracle Lite supports three types of refresh: fast refresh, complete refresh, and optimum refresh.

Fast Refresh

A fast refresh uses the snapshot transaction logs at the master site to refresh a simple snapshot by transmitting only the changes needed to bring the snapshot up to date. Only simple snapshots can execute a fast refresh. Generally a simple snapshot should

use fast refresh because it is the most efficient refresh type. If you want to use fast refresh, you must create a snapshot log for the table at the master site. For instructions, see Chapter 2, “Preparing for Replication.”

During a fast refresh, the following events occur in sequence:

1. The ROWID or primary key entries logged in the snapshot transaction logs at the master site are selected and transferred to the snapshot site.
2. Rows deleted from the master table are deleted at the snapshot table.
3. Updates to rows at the master table are applied at the snapshot table.
4. Any new rows inserted at the master table are inserted in the snapshot table.

Complete Refresh

A complete refresh entirely replaces the existing data in a simple or complex snapshot. All existing rows in the snapshot table are first deleted, then all rows of the master table that are qualified by the snapshot query are transferred to the client and inserted into the empty snapshot table.

Optimum Refresh

An optimum refresh will attempt to execute a fast refresh, if possible. If not possible, such as in the case of a complex snapshot or a simple snapshot without a snapshot log, it will execute a complete refresh.

Refresh Group

Personal Oracle Lite uses a refresh group to specify which snapshots a refresh action will affect. There is only one refresh group. An application can define the refresh group once to specify all snapshots and refresh them all every time, or it can redefine the group with a different collection of snapshots each time.

Normally, all snapshots should be added to the refresh group. If you do not add a snapshot to the refresh group, it is not updated with master site changes during a refresh. However, there might be cases where you would want to push changes from the snapshot but not pull changes from the master table. In such cases, you would not add the snapshot to the refresh group.

Error Conditions During Refresh

When a push error (caused by conflict or constraint violation) is encountered during the refresh process, the error condition decides whether it should abort or proceed with the operation. There are two error conditions: Stop On Error and Continue On Error.

Stop On Error

With the Stop-on-Error condition, a push error is returned and the refresh process is aborted. Since the refresh is aborted, the snapshot is not updated with the data in the master table.

Continue On Error

With the Continue-on-Error condition, no push error is returned and the refresh process proceeds even though there is a push error. Since the invalid transaction is not applied to the master table, the invalid records in the snapshot before the refresh will not reappear in the snapshot after the refresh.

Refresh-In-Progress Mode

When an application performs a refresh operation, the connected snapshot site enters the Refresh-In-Progress mode. During this mode, any attempts to update any updatable snapshots at that snapshot site will be rejected. The snapshot site will return back to the normal mode when the refresh operation finishes. In some cases, the snapshot site automatically restores back to the normal mode without a complete refresh; for example, the machine is powered off or the application goes down during the Refresh-In-Progress mode. When the application reconnects, the snapshot site is returned to the normal mode.

Conflict Detection and Resolution

In general, the master site handles conflict detection and resolution for the snapshot site. However, you can minimize the potential for conflicts by understanding the conflict detection and resolution strategy at the master site. For example, an application that uses updatable snapshots should not allow changes to primary key values in a snapshot row. This is because the master site uses the primary key during conflict detection to determine which rows are compared.

You may want to consult with the database administrator of the master site about these and other rules. For more information about Oracle conflict resolution methods, refer to the documentation for your master site: *Oracle7 Server Distributed Systems, Volume II: Replicated Data*.

When replicating with Personal Oracle Lite, three types of conflicts can be detected by the master site:

- update conflicts

- uniqueness conflicts
- delete conflicts

The procedures at the master site detect an **update conflict** if there is any difference between the old values of the replicated row and the current values of the same row at the master site. A snapshot site sends new row values to update the corresponding row values at the master site. The snapshot site also sends the old row values for comparison with the current values at the master site. If the old row values match the current master row values, the rows at the master site are assigned the new values. But if the old row values do not match the current master row values, an update conflict is detected.

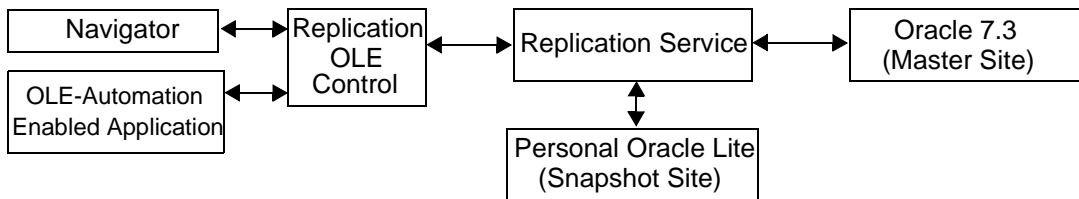
A **uniqueness conflict** is detected if a unique constraint is violated during an INSERT or UPDATE of the replicated row.

A **delete conflict** is detected when a row has been deleted from an updatable snapshot, and the old values of the deleted row do not match the current values at the master site. For example, if a row at the master site is changed after the corresponding row is deleted at the snapshot site, a delete conflict is detected.

Replication OLE Control (OCX)

The Personal Oracle Lite Replication OLE Control acts as an interface for replication operations done either directly or programmatically. Application developers using an OLE-Automation enabled application development tool such as Visual Basic can programmatically interact with the Replication OLE control through its methods. For more information, see Appendix B, “Replication OLE Control Methods.”

The diagram below shows how Personal Oracle Lite replication works with either the Navigator or an application created in an OLE-Automation enabled development tool.



You can either embed the Replication OLE control in an OLE container, or you can create it at runtime (for example, using CreateObject() in Visual Basic). The ProgID (Programmatic Identifier) for creating a replication object is `Polite.RepSvr`. Before

doing any operation (like creating a snapshot), connections to both a master site and the snapshot site must be opened successfully.

When an application is deployed, an end user performs replication by clicking on a simple user interface control provided by the application.

2

Preparing for Replication

To use Personal Oracle Lite replication, both the master site and the snapshot site must be configured correctly to support replication operations. This chapter describes:

- preparing the master site for replication
- preparing the snapshot site for replication
- setting up replication connectivity

Preparing the Master Site for Replication

In order to ensure that your applications can replicate data successfully with Personal Oracle Lite, you must prepare the master site for replication. The steps required to prepare the master site are as follows:

- verifying Oracle7 server support for replication
- setting up support for the replication engine
- creating snapshot transaction logs (if using fast refresh)
- preparing master tables for replication

Verify Oracle7 Server Support for Replication

Replication applications for use with Personal Oracle Lite require a master site that supports full replication services. The database administrator of the master site should verify that the database is an Oracle7 Universal Server Enterprise Edition running version 7.3 or later, and that replication support is active and configured properly. The Oracle7 Workgroup Server is not suitable, because it only supports read-only snapshot replication to Personal Oracle Lite databases.

Set Up Support for the Replication Engine

The Personal Oracle Lite replication engine (REPAPI) facilitates replication between Personal Oracle Lite and Oracle7 databases. Part of preparing the master site for replication requires executing a SQL script that installs a table on the master site to support the Personal Oracle Lite replication engine.

The database administrator of the master site will need to run this script, since administrator privileges are required to write to the files the script uses. The script is called REP73.SQL, and is provided with Personal Oracle Lite, in the installed directory:

Oracle_Home\dbs

where *Oracle_Home* is:

- ORAWIN95 on Windows 95 systems
- ORANT on Windows NT systems
- ORAWIN on Windows 3.1x systems

Create Snapshot Logs (for Fast Refresh)

If you plan to use fast refresh (recommended), you need to create a snapshot log at the master site for each master table to be replicated. To create a snapshot log for a master table, issue the following SQL command:

```
create snapshot log on schema.mastertablename;
```

where `schema` and `tablename` are the schema and name of the master table to be replicated. This will create a table called `MLOG$_tablename` at the master site, containing the snapshot log.

Prepare Master Tables for Replication

You will need to define the group of master tables at the master site that you want to replicate. You can do this using the Oracle Replication Manager utility or another, similar tool. At a minimum, you need to create a master group and include in it all master tables you want to replicate. You must also assign primary keys to all master tables in the group. For more information, refer to the documentation for your replication administration tool.

Preparing the Snapshot Site for Replication

When a snapshot is first created, Personal Oracle Lite queries the master table and automatically creates a **base table** with the same columns and data definitions as the master table. It also includes an extra column, `M_ROW$$`, as the last column of the base table. This extra column is used to identify the rows in the master table. After the base table is created, Personal Oracle Lite stores the snapshot data in it.

When creating a snapshot through an application or the Navigator, no data integrity constraints that exist on the master table are duplicated in the snapshot base table except for the NOT NULL constraint. If a snapshot needs to have data integrity constraints, you need to do one of the following:

- add constraints to the base table after you create the snapshot. To add constraints, use the SQL command, `ALTER TABLE`, or
- create its base table at the snapshot site *before* you create the snapshot through your application or the Navigator. This process is known as **pre-creating** the snapshot base table, and is described below.

Pre-Creating Snapshot Base Tables

If you are creating a new snapshot that requires integrity constraints or adding integrity constraints to an existing snapshot, you can create the base table with the new integrity constraints, initially.

Note: In general, it is good practice to place the same integrity constraints on the snapshot that exist on the master table.

To pre-create a snapshot base table:

1. Use the SQL command `CREATE TABLE`, and specify the name of the snapshot.
2. Specify the same column definitions and constraints as the master table.
3. Specify an extra column, `M_ROW$$`, with datatype `VARCHAR2 (18)`, as the last column of the table

The data definition (number of columns and their datatypes) of pre-created snapshot base tables must match the data definition of the master tables to be replicated. If the data definition of a master table changes, but the base table derived from it has not been recreated to match the new definition, refreshing the snapshot will not succeed.

For example, to use a snapshot called `foo` of a master table, first create a base table called `foo` at the snapshot site. The base table should have the same data definition and integrity constraints as the master table.

The base table definition should also have an additional column called `M_ROW$$` with a datatype of `VARCHAR2 (18)`. This column contains the ROWIDs of the master table. Once the snapshot associated with the base table is created, the `M_ROW$$` column becomes implicit. In other words, the column is not returned by queries on the snapshot table unless it is specified explicitly in the query. For example, the following select statement does not return values from `M_ROW$$`:

```
select * from foo;
```

while the following select statement does return values from `M_ROW$$`:

```
select c1, c2, ... , m_row$$ from foo;
```

After you create the snapshot base table with any necessary integrity constraints, you can create its associated snapshot using the Navigator or from within your application using the Replication OLE control method `SnapshotStoreCreate()`.

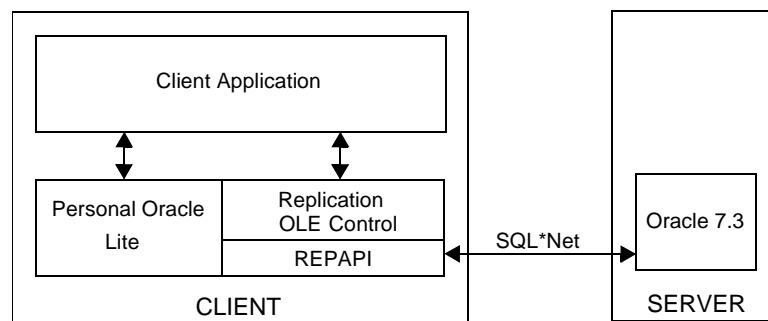
Setting Up Replication Connectivity

Personal Oracle Lite supports both store-and-forward (asynchronous) replication with Oracle Mobile Agents and real-time (synchronous) replication with SQL*Net.

Whether to use SQL*Net or Oracle Mobile Agents connectivity will depend on your application and business needs. For wireless communications, Oracle Mobile Agents is the only option. For dial-up and LAN communications, either method can be used.

SQL*Net Replication

SQL*Net is Oracle's remote data access software enabling client-server communications across a network. With SQL*Net, the basic send and receive requests between client and server are *synchronous*. That is, when the client initiates a request, it waits for the server to respond with the answer. It can then issue an additional request.



Personal Oracle Lite Synchronous Replication with SQL*Net

The following are some issues to consider when determining whether to use SQL*Net replication:

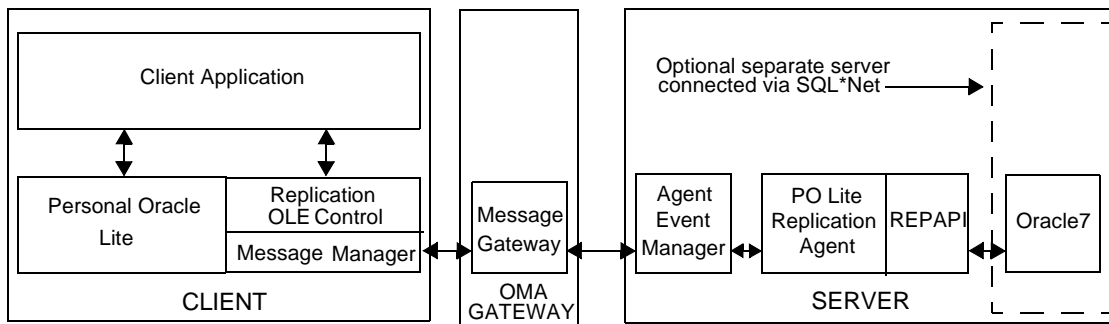
- **Cost** SQL*Net software is installed automatically when you install Personal Oracle Lite, and requires minimal configuration. It is also freely distributable.
- **Complexity** SQL*Net is easy to install and configure. In addition, with synchronous replication, your application does not have to handle notification.
- **Footprint** In the SQL*Net model, the Personal Oracle Lite replication engine (REPAPI) and its support files are installed on the client machine. This can mean a larger footprint on the client.
- **Availability** SQL*Net is available on many platforms.

- **Performance** Across a LAN, SQL*Net is comparable in performance to Oracle Mobile Agents

To install SQL*Net, follow the instructions in your *Personal Oracle Lite Installation Guide* CD insert. Once installed, you can configure SQL*Net by double-clicking the SQL*Net Easy Configuration icon in your Personal Oracle Lite program group. For more information, refer to the SQL*Net online help.

Oracle Mobile Agents Replication

Oracle Mobile Agents is a networking communications middleware that enables bi-directional data replication over LAN, dial-up, and wireless networks. With Oracle Mobile Agents, the basic send and receive requests between client and server are **asynchronous**. That is, when the client initiates a request, it does not have to wait for the server to respond with the answer. It can disconnect or continue working. Client components include a Message Manager and Message Manager configuration tool. The server components include a Message Gateway and Agent Event Manager which can be installed on different machines.



Personal Oracle Lite Asynchronous Replication with Oracle Mobile Agents

The following are some issues to consider when determining whether to use Oracle Mobile Agents replication:

- **Cost** The client software for Oracle Mobile Agents is distributable for a licensing fee.
- **Complexity** Oracle Mobile Agents requires installation, configuration, and administration of the client, gateway, and agent software. In addition, with asynchronous replication, the application needs to be designed to handle notification.

- **Footprint** In the Oracle Mobile Agents model, the Personal Oracle Lite replication engine (REPAPI) is installed on the server. This can mean a smaller footprint on the client machine.
- **Availability** Components of Oracle Mobile Agents are available on a limited number of platforms. In addition, the Personal Oracle Lite Replication agent currently is only available for Windows NT servers.
- **Performance** Dial-up performance over PPP is significantly faster.

To install Oracle Mobile Agents, follow the instructions in your *Personal Oracle Lite Installation Guide* CD insert. Once installed, you can configure Oracle Mobile Agents by following the instructions in Appendix A, “Setting Up Oracle Mobile Agents.” For more information, refer to the Oracle Mobile Agents online documentation.

Sample Application

A sample Visual Basic 4.0 application that demonstrates replication with Oracle Mobile Agents is included with Personal Oracle Lite. The name of the application is OTREP16.MAK on Windows 3.1x, and OTREP32.MAK on Windows 95 and Windows NT. The application is located in your *Oracle_Home*\Lite\Examples\REPSVR directory. For a description of this application, see Appendix C, “Sample Replication Application.”

3

Using Replication

This chapter describes how to perform replication operations from within an application and using the Navigator. Specifically, it describes:

- connecting to databases
- working with snapshots
- scheduling refreshes
- handling errors during refresh
- working with Oracle Mobile Agents
- deploying replication applications
- running a sample application
- troubleshooting common problems

Before you can use replication, you must properly prepare the master and snapshot sites. For more information, see Chapter 2, “Preparing for Replication.”

Overview

Personal Oracle Lite provides two ways to perform replication tasks:

- Manually, using the Navigator

The Navigator is a tool that provides a graphical, point-and-click interface for managing database objects and performing replication tasks with Personal Oracle Lite. The Navigator is available only for 32-bit operating system environments. For more information about the Navigator, refer to the online help available with the tool.

- Programmatically, through an application created with an OLE-Automation enabled application development tool, such as Visual Basic.

This chapter assumes your application is able to take advantage of Personal Oracle Lite's Replication OLE control methods. For more information about the Replication OLE control, see Appendix B, "Replication OLE Control Methods."

Connecting to Databases

Before a program can perform replication operations in Personal Oracle Lite, it must first establish a connection to the master site, and then connect to the snapshot site.

Note: These instructions assume that you have already created and configured your database connections. For more information, see the Navigator online help.

To open a connection from within an application, use the Replication OLE control methods `MasterConnectionOpen()` to connect to the master site and `StoreConnectionOpen()` to connect to the snapshot site.

To open a connection to the master site using the Navigator.

1. Double-click on the Database Connections folder.
2. Double-click on the master site database node to open the connection.

To open a connection to the snapshot site using the Navigator.

1. Double-click on the Oracle Lite Databases folder.
2. Double-click on the snapshot site database node to open the connection.

Working with Snapshots

After establishing a connection to the master site and snapshot site, you can perform the following operations on snapshots:

- creating snapshots
- adding snapshots to the snapshot refresh group
- refreshing snapshots
- dropping snapshots

These operations are described in the following sections.

Creating Snapshots

If a snapshot will have data integrity constraints, the base table that will contain it can be created first. See Chapter 2, “Pre-Creating Snapshot Base Tables.”

To create a snapshot from within an application, use the `SnapshotStoreCreate()` method. For more information, see Appendix B, “Replication OLE Control Methods.”

To create a snapshot using the Navigator:

1. Connect to the master site (see “Connecting to Databases,” earlier in this chapter).
2. Locate the icon for the master table you wish to replicate.
3. Connect to the snapshot site (see “Connecting to Databases,” earlier in this chapter).
4. Locate the Snapshots folder.
5. Create the snapshot using either of these two methods:
 - Drag a copy of the master table and drop it into the Snapshots folder.
 - Select the master table and choose Copy from the right-mouse menu. Then select the Snapshots folder and choose Paste from the right-mouse menu.
6. In the Create Snapshot dialog, specify the type of snapshot you want to create, either read-only or updatable.
7. In the New Snapshot property sheet, select the General tab. Specify a name for the snapshot, and the owner.

Optionally, you can specify a WHERE clause to qualify the query that produces the snapshot data from the master table. Select the Condition tab, and in the entry field, type the condition or select keywords from the selection lists.

When you are finished, click OK. The snapshot is created in the Snapshots folder

Adding Snapshots to the Snapshot Refresh Group

You can add a snapshot to the snapshot refresh group from within an application, using the Replication OLE control method `SnapshotGroupAdd()`. The Navigator does not support this feature. All snapshots you create in the Navigator are added to the snapshot refresh group automatically.

Refreshing Snapshots

Refreshing updatable snapshots is more complex than refreshing read-only snapshots because updates at both the master site and the snapshot site must be considered. All transactions performed on updatable snapshots are pushed up to the master site, but only the snapshots in the refresh group will be updated with new or changed information from the master site.

To refresh snapshots from within an application, use the Replication OLE control methods in the sequence listed below:

1. Reset the refresh group to contain no snapshots using `SnapshotGroupReset()`
2. Add the snapshots to be refreshed to the group using `SnapshotGroupAdd()`
3. Refresh the snapshot group using `SnapshotGroupRefresh()`

To refresh a snapshot from the Navigator, select the snapshot and choose Refresh from the right-mouse menu. This procedure causes Personal Oracle Lite to push changes to the master site for all updatable snapshots, and to pull changes only for the selected snapshot. To update all the snapshots using Navigator, refresh each one explicitly.

Dropping Snapshots

You can drop, or delete, a snapshot when it is no longer needed.

To drop a snapshot from within an application, use the Replication OLE control method `SnapshotStoreDrop()`.

To drop a snapshot using the Navigator, select the snapshot and select Delete from the right-mouse menu.

Scheduling Refreshes

An application can refresh snapshots at different intervals. It can refresh a snapshot every time a connection is established with the master site, or less frequently, such as once a day, once per week or longer. Consider the following when deciding when to do a refresh:

- If the master tables receive predictable updates at regular intervals, refresh the associated snapshots at the appropriate intervals.
- After bulk loads to the master tables, refresh all snapshots based on the master tables. This propagates the new rows of the master tables to associated snapshots.
- Since larger snapshots take longer to refresh, consider refreshing them less frequently.
- If a snapshot is based on data that is not prone to change, refresh the snapshot less frequently.

Handling Errors During Refresh

The snapshot refresh process may fail if error conditions or conflicts occur at the master or snapshot sites. If the push fails, the status of the pending transaction is stored in an error queue, DEF\$_ERROR, at the master site.

Once the error has been corrected or the conflict resolved, the pending transaction instructions are applied in the correct order so the operation can complete successfully.

To handle errors from within an application, you can use the Replication OLE control method `ErrorInfoGet()` to retrieve the text of the error messages for any method that returns a `FALSE` if it fails to execute.

In the Navigator, the default error condition is Continue on Error, which cannot be changed.

Working with Oracle Mobile Agents

This section describes two features specific to replication with Oracle Mobile Agents:

- handling notification
- using the long transaction feature

Handling Notification

With Oracle Mobile Agents replication, Personal Oracle Lite simulates a pushbutton click event to notify an application when a pending operation is completed. In order to receive the notification, an application must implement an event handler for the pushbutton click event. This is required for all replication applications using Oracle Mobile Agents.

To implement an event handler from within an application, you must use the Replication OLE control method `Initialize()`. This method requires you to provide a window handle for notification, a pushbutton control ID, and an Oracle Mobile Agents Network Name.

Using the Long Transaction Feature

Long Transaction is an advanced feature available only with Oracle Mobile Agents replication. Normally, an Oracle Mobile Agents replication application can disconnect from the snapshot site and exit before a replication operation is completed. By default, this causes the operation to be aborted, and the application must resubmit the operation when it reconnects again. With Long Transaction enabled, an application can disconnect from the snapshot site before a replication operation is completed, and reconnect again to either resume or abort the pending operation. If the operation is resumed, it will be completed normally.

To enable Long Transaction from within an application, you must use the Replication OLE control methods `Initialize()`, `LongTransactionEnable()`, `ReplicationProgressGet()`, `ReplicationResume()`, and `ReplicationAbort()`.

When an application uses the above methods, it is said to be “long-transaction aware.” For an example of a long-transaction aware application, see Appendix C, “Sample Replication Application.”

The following rules apply when using Long Transaction:

- On a single machine, only one application can perform replication operations at a time. Any attempts to access replication service concurrently from other applications will return an error.
- On a single machine, only one snapshot site can be in long transaction mode at a time.
- When a replication application connects to a snapshot site in long transaction mode, it can either resume or abort any pending replication operation.
- When a replication application that is *not* long-transaction aware attempts to perform replication in long transaction mode, it automatically aborts any pending replication operation and causes the snapshot site to return to the normal mode.

Deploying Replication Applications

This section describes the steps required to deploy Personal Oracle Lite replication applications, including:

- distributing required software
- clearing snapshot transaction logs (if distributing a starter database)
- recreating snapshots (if distributing a starter database and using fast refresh)

Distribute the Required Software

The software you need to include when deploying a Personal Oracle Lite replication application is listed in the table below.

Note: The file locations are on the Personal Oracle Lite CD-ROM, unless otherwise noted

Software to include:	Where located:
Personal Oracle Lite Runtime	For the complete list of files, refer to the document DEPLOY.TXT in your installed <i>Oracle_Home</i> \LITE directory.
Replication OLE control	\POL23\POLREP\OTSVRxx.OCX

SQL*Net (if using SQL*Net replication)	SQL*Net files: \WINDOWS\SQLNET*. * or \WIN32\SQLNET*. * and REPAPI files: \WINDOWS\RSF73*. * or \WIN32\V7\RSF73*. *
Oracle Mobile Agents (if using Oracle Mobile Agents replication)	Oracle Mobile Agents client software: \OMA\CLIENT*. * and Oracle Mobile Agents client-side Message Manager software: \OMA\CLIENT\<platform>\OMAC *. * (See the Oracle Mobile Agents online documentation for how to “auto-install” using the Oracle Installer.) and Personal Oracle Lite client component for Oracle Mobile Agents: \WIN32\POL23\POLOMA\POLOMAxx.DLL

Clearing Snapshot Transaction Logs

If you are distributing a starter database that has snapshots already created, you will need to clear its snapshot transaction logs:

- delete all rows from USLOG
- delete all rows from USLOG\$_*snapshotname*

If you deploy a database with transactions still recorded in the logs, it will fail to perform replication.

Recreate Snapshots

If you are distributing a starter database and your application uses fast refresh, you will need to recreate snapshots, rather than simply copy them, when your application installs the database on the client machine.

Fast refresh depends on logs at the master site that enable it to refresh only the data that has changed. The snapshot logs at the master site (SLOG and MLOG\$_*tablename*) keep track of snapshots by their unique ID numbers. When a snapshot is first created, it is given a unique ID number based on the system timestamp. This enables the master

site to distinguish between snapshots at different snapshot sites that are based on the same master table.

If your application simply copies a starter database to multiple machines, all snapshots based on the same master table will share the same ID number. For example, snapshot DEPT on machine A will have the same ID number as snapshot DEPT on machine B. This can cause the master site to ignore what may appear to be duplicate transactions based on prior entries in the logs.

Thus, if your application uses fast refresh, you need to recreate snapshots during installation, to ensure that a unique ID number for each snapshot is assigned.

Running a Sample Application

A sample Visual Basic 4.0 application that demonstrates replication with Oracle Mobile Agents is included with Personal Oracle Lite. The name of the application is OTREP16.MAK on Windows 3.1x, and OTREP32.MAK on Windows 95 and Windows NT. The application is located in your *Oracle_Home\Lite\Examples\REPSVR* directory. For a description of this application, see Appendix C, “Sample Replication Application.”

Troubleshooting

This section describes common replication problems and solutions.

Using the Navigator, I am able to refresh my updatable snapshots without error, yet the changes are being lost. Why?

It may be possible that the master tables have not been set up properly for replication, and your snapshots are invalid. Unfortunately, the Navigator has no way to verify this, and will permit you to perform replication tasks without error. If you suspect the master tables were not prepared properly for replication, you will need the database administrator of the master site to assist you. For more information, see Chapter 2, “Preparing for Replication.”

If you have properly prepared for replication and still are not seeing your changes, it may be that your transaction caused a conflict. Suppose your transaction causes a conflict by violating a primary key. The invalid transaction is recorded in an error

queue at the master site instead of being applied to the master table. Since the Navigator is always set to Continue on Error, you do not receive an error message. When the changes are pulled down to the snapshot, they appear to be lost. However, the invalid transaction is not lost, it is just recorded in the error queue. The database administrator of the master site can help you by checking the error queue and working with you to resolve the conflict and re-execute the transaction.

Why are my simple snapshots taking so long to refresh?

You may be using a complete refresh for simple snapshots, when you can use a fast refresh, instead. A complete refresh entirely deletes and recreates the data in a snapshot, whereas a fast refresh transmits only the changes needed to bring the snapshot up to date. Only a simple snapshot (consisting of a full table or subset of rows) can have a snapshot log at the master site that makes it possible to execute a fast refresh.

For simple snapshots with snapshot logs, you should use fast refresh because it is more efficient. If you have both simple and complex snapshots, consider using optimum refresh. With optimum refresh, the snapshot will execute a fast refresh whenever possible.

My application executes a long, complex query against a very large database. What can I do to speed up refresh time?

You may be able to improve refresh time by increasing the initial size of the object cache. You can specify a larger cache size by using the CacheSize parameter in your POLITE.INI file. For more information on POLITE.INI, see Appendix B, “Database Parameters in POLITE.INI,” in the *Personal Oracle Lite Programmer’s Guide*.

Suppose I open a connection to my snapshot site. Then I open a second connection and create a snapshot. If I go back to the first connection and insert data in the snapshot, this data is lost when I refresh the snapshot. Why?

Because of the way Personal Oracle Lite internal triggers create entries in snapshot logs when you change the snapshot data, the connection in which the snapshot is created, and any connection opened thereafter will record changes in the transaction logs. However, any connection that existed before the snapshot was created will *not* record changes in the transaction logs. As a result, changes can be lost when you refresh the snapshot.

To ensure data integrity when using multiple connections, close and reopen all connections after creating snapshots. Closing and opening the connections will enable the underlying triggers that control updates to the logs, ensuring that no connection is pre-existing to the snapshot, and the logs are correctly updated.

A

Setting Up Oracle Mobile Agents

This section provides a summary of the steps necessary to configure Oracle Mobile Agents software on the client and server in order to use Personal Oracle Lite replication with Oracle Mobile Agents.

Note: These instructions assume that you have already installed the Oracle Mobile Agents Replication option, and the Oracle Mobile Agents client and server components. For more information, refer to the *Personal Oracle Lite Installation Guide* CD insert.

For more information on Oracle Mobile Agents, refer to the Oracle Mobile Agents online documentation, available in Adobe Acrobat (.PDF) format on the Personal Oracle Lite CD-ROM. The following online documents are located in the \OMA directory:

- Oracle Mobile Agents User's Guide (OMAU.SR.PDF)
- Oracle Mobile Agents Administrator's Guide (OMAADM.PDF)

Configuring the Oracle Mobile Agents Client Software

1. To configure the Message Manager, double-click the Oracle Configuration icon in the Oracle Mobile Agents program group. In the Oracle Mobile Agents Configuration dialog, fill in the information to configure the Message Manager.
 - You can use the default network, called “Home Network,” or you can create a new network. To create a new network, choose Network from the Advanced menu. If you are unsure about your network information, contact your System Administrator.
 - Fill in the username and password information for the network.
 - Under Configure Drivers, use the buttons to toggle between the types of drivers you want to configure. The configuration information for the selected driver is displayed. If you are unsure about your configuration information, contact your System Administrator.
 - Choose Edit from the Advanced/Services menu, to register a service for Personal Oracle Lite replication. In the New Services dialog, type: `poloma.services.mobile.oracle.com`. Then click OK. In the Edit Service Information dialog, click Save. The service will display as “Registered” after you have run the Message Manager (see step 2, below).
 - When you are finished entering configuration information, click Save to accept the Oracle Mobile Agents Configuration dialog.
2. To run the Message Manager, double-click the Oracle Mobile Agents icon in the Oracle Mobile Agents program group. You must keep this window open while using Oracle Mobile Agents, but you can minimize it by clicking Minimize.

Configuring the Oracle Mobile Agents Server Software (for Database Administrators)

1. To configure agents, double-click the Agent Control icon in the Oracle Mobile Agents program group. In the Agent Control dialog, fill in the information to configure agents.
 - To initialize the primary Personal Oracle Lite replication agent, choose Initialize from the Agent menu. From the Open file dialog, select `poloma.exe` and click OK. In the Install New Agent dialog, `poloma` appears as the new agent. Click OK to install the agent and return to the Agent Control dialog.
 - To configure the Personal Oracle Lite replication agent, `poloma`, select it from the list of agents.

- Click the Network tab, to specify gateway information for the agent. If you do not know the host name, contact your System Administrator.
- Click the Parameters tab, to specify parameters for the agent. The minimum parameters you must create are “access” and “share_path.”

To create the “access” parameter: In the Parameter field, type `access`. Tab to the Value field and type `write`. Then click Add.

To create the “share_path” parameter: In the Parameter field, type `share_path`. Tab to the Value field and type the path for the agent. For example, the path to your primary agent might be:

`d:\Oracle_Home\mobile\agents\poloma.`

Then click Add.

- To create additional Personal Oracle Lite replication agents, copy the primary agent executable file to a new file. For example, copy “poloma.exe” in your *Oracle_Home*\bin directory and name the new file “poloma1.exe.” Then repeat step 1 to initialize and configure the new agent.
 - When you have initialized and configured the agent, click Start to start the agent.
2. To start and stop the Message Gateway and check its status, double-click the Message Gateway Control icon in the Oracle Mobile Agents program group. Once started, the Gateway Status should be “running.” If you click Start and the message gateway is still not running, contact your System Administrator.

B

Replication OLE Control Methods

The methods described in this section perform the following operations:

- opening a connection to the master site
- opening a connection to the snapshot site
- initializing replication service
- creating a snapshot
- resetting the refresh group
- adding snapshots to the refresh group
- refreshing the refresh group
- dropping a snapshot
- retrieving error information
- closing the connection to the master site
- closing the connection to the snapshot site
- enabling long transaction mode
- getting the long transaction status
- resuming the replication operation
- aborting the replication operation

Opening a Connection to the Master Site

Method: MasterConnectionOpen(String Username, String Password, String Conn_str)

Parameters: Username Name of the user on the Oracle master database.
Password Password for the user.
Conn_str SQL*Net version 2 connect string.

Returns: TRUE Success.
FALSE Fail.

Examples: MasterConnectionOpen('SCOTT', 'TIGER', 'MASTERDB')

Opening a Connection to the Snapshot Site

Method: StoreConnectionOpen(String Username, String Password, String Conn_str)

Parameters: Username Name of the user on Personal Oracle Lite.
Password Password for the user (currently ignored).
Conn_str Data source name and an optional database name.

Returns: TRUE Success.
FALSE Fail.

Examples: StoreConnectionOpen('OOT_SCH', 'OOT_SCH', 'POLITE')

Note: To connect to a particular database, for example, MYDB, the Conn_str must be POLITE:MYDB.

Initializing Replication Service (required, for Oracle Mobile Agents)

Method: Initialize(Integer nMode, String Hwnd, String Control ID, String Network Name)

Parameters: nMode Indicates type of replication connection.

Value:	Description:
0	SQL*Net
1	Oracle Mobile Agents

Defaults to 0.

Hwnd Window handle to receive notification (required, if nMode is 1).

Control ID Pushbutton control ID to serve notification (required, if nMode is 1).

Network Name Oracle Mobile Agents Network Name (required, if *nMode* is 1).

Returns:

TRUE	Success.
FALSE	Fail.

Examples: `Initialize(1, RepMain.hwnd, 100, 'OMA-Network')`

Creating a Snapshot

Method: `SnapshotStoreCreate(String Store_owner, String Store_name, String Master_owner, String Master_table, String Query, Bool Updatable, Integer Options)`

Parameters:

<i>Store_owner</i>	Owner of the client snapshot. Must be upper case.
<i>Store_name</i>	Name of the snapshot. Must be upper case.
<i>Master_owner</i>	Owner name of the master table.
<i>Master_table</i>	Name of the master table.
<i>Query</i>	SQL query for the snapshot.
<i>Updatable</i>	Indicates if the snapshot is updatable.
	Value: Description:
	TRUE Updatable
	FALSE Read-only
<i>Options</i>	Indicates if the replication is simple or complex.
	Value: Description:
	0 Simple
	1 Complex

A replication is complex if the SQL query contains subqueries, joins, set operations, or any of the following keywords or clauses:

- DISTINCT
- AGGREGATE
- GROUP BY
- CONNECT BY

Returns:

TRUE	Success.
------	----------

FALSE Fail.

Examples:

```
SnapshotStoreCreate('OOT_SCH', 'EMP', 'SCOTT', 'EMP',
                    'SELECT * FROM SCOTT.EMP',
                    TRUE, 0)
```

The following three methods must be called in sequence to execute a complete snapshot refresh operation: `SnapshotGroupReset()`, `SnapshotGroupAdd()`, and `SnapshotGroupRefresh()`.

Resetting the Refresh Group

This operation cleans up the previous refresh snapshot list in the refresh group.

Method: `SnapshotGroupReset()`

Parameters: None.

Returns: None.

Adding Snapshots to the Refresh Group

Method: `SnapshotGroupAdd(String Store_owner, String Store_name)`

Parameters:

<i>Store_owner</i>	Owner of the client snapshot. Must be upper case.
--------------------	---

<i>Store_name</i>	Name of the snapshot. Must be upper case.
-------------------	---

Returns: None.

Examples: `SnapshotGroupAdd('OOT_SCH', 'EMP')`

Refreshing the Refresh Group

Note: Calling this method alone does not constitute a complete snapshot refresh operation. To refresh snapshots, you must call three methods in sequence: `SnapshotGroupReset()`, `SnapshotGroupAdd()`, and `SnapshotGroupRefresh()`.

Method: `SnapshotGroupRefresh(Int Refreshmode, Bool StopOnError)`

Parameters:

<i>Refreshmode</i>	Indicates how client data store is refreshed.
--------------------	---

Value:	Description:
--------	--------------

0	Fast refresh.
---	---------------

	1	Complete refresh.
	2	Optimum refresh.
<i>StopOnError</i>	Indicates whether to continue a refresh when errors occur.	
	Value:	Description:
	TRUE	Stop refresh.
	FALSE	Continues refresh.

Note: This flag only applies to updatable snapshots.

Returns:

TRUE	Success.
FALSE	Fail.

Examples: `SnapshotGroupRefresh(2, TRUE)`

Dropping a Snapshot

Method: `SnapshotStoreDrop(String Store_owner, String Store_name)`

Parameters:

<i>Store_owner</i>	Owner of the client snapshot. Must be upper case.
<i>Store_name</i>	Name of the snapshot. Must be upper case.

Returns:

TRUE	Success.
FALSE	Fail.

Examples: `SnapshotStoreDrop('OOT_SCH', 'EMP')`

Retrieving Error Information

Method: `ErrorInfoGet()`

Parameters: None.

Returns: String containing all replication error messages encountered for the last method used. Messages are separated by a new line.

Examples: `ErrorInfoGet()`

Closing the Connection to the Master Site

Method: `MasterConnectionClose()`

Parameters: None.

Returns:	TRUE	Success.
	FALSE	Fail.

Closing the Connection to the Snapshot Site

Method:	StoreConnectionClose()	
Parameters:	None.	
Returns:	TRUE	Success.
	FALSE	Fail.

Enabling Long Transaction Mode (Advanced Feature)

Method:	LongTransactionEnable(Bool LongTransaction)	
Parameters:	LongTransactionIndicates whether to enable long transaction mode.	
	Value:	Description:
	TRUE	Enables long transaction mode.
	FALSE	Does not enable long transaction mode.
Returns:	TRUE	Success.
	FALSE	Fail.

Getting the Long Transaction Status (Advanced Feature)

Method:	ReplicationProgressGet()	
Parameters:	None.	
Returns:	TRUE	The connected snapshot site is in long transaction mode.
	FALSE	The connected snapshot site is not in long transaction mode.

Resuming the Replication Operation (Advanced Feature)

Method:	ReplicationResume()	
Parameters:	None.	
Returns:	TRUE	Success.
	FALSE	Fail.

Aborting the Replication Operation (Advanced Feature)

Method: `ReplicationAbort()`

Parameters: None.

Returns: `TRUE` Success.
`FALSE` Fail.

C

Sample Replication Application

The purpose of this chapter is to familiarize you with Personal Oracle Lite replication tasks and their corresponding Replication OLE control methods. It describes a Visual Basic application that demonstrates replication with Oracle Mobile Agents. The application is called OTREP16.MAK on Windows 3.1x, and OTREP32.MAK on Windows 95 and Windows NT, and is located in your *Oracle_Home*\Lite\Examples\REPSVR directory.

About this Application

The sample Visual Basic application described below performs common replication operations such as connecting to the master and snapshot sites, initializing the replication service for Oracle Mobile Agents, and creating, refreshing, adding, and dropping snapshots. It also demonstrates the long transaction, an advanced feature available with Oracle Mobile Agents replication.

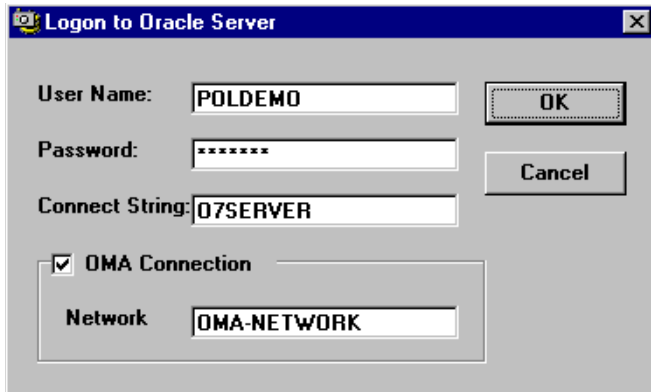
The forms contained in this application and their associated actions are described below:

Form Name:	Associated Actions:
Logon to Oracle Server (repmsr.frm)	1. Connect to the master site 1.1. Check whether OMA Connection is selected 1.2. Initialize replication service (if using OMA) 1.3. Connect to the master site
Logon to Personal Oracle Lite (repstr.frm)	2. Connect to the snapshot site
Personal Oracle Lite Replication (repmain.frm)	3. Perform replication operations 3.1. Initiate an instance of Replication OLE control 3.2. Create a snapshot (calls Create Snapshot form) 3.3. Reset the snapshot refresh group 3.4. Add a snapshot (calls Add Snapshot form) 3.5. Refresh the snapshot refresh group 3.6. Get error information 3.7. Drop a snapshot (calls Drop Snapshot form) 3.8. Handle notification (if using OMA) 3.9. Enable the long transaction feature 3.10. Get long transaction status 3.11. Resume the pending operation 3.12. Abort the pending operation
Create New Snapshot (repctr.frm)	4. Create a snapshot
Add/Drop Snapshot (repadd.frm)	5. Add or drop a snapshot 5.1. Add a snapshot to the refresh group 5.2. Drop a snapshot

1. Connecting to the Master Site

Use this form and the associated code to perform the following actions when connecting to the master site (Oracle remote database):

- 1.1. Check whether OMA Connection is selected
- 1.2. Initialize replication service (if OMA Connection is selected)
- 1.3. Connect to the master site



1.1. Check whether OMA Connection is selected

```
'Check for OMA Connection
g_bOMA = OMAMessage
```

1.2. Initialize replication service, if OMA Connection is selected

```
If g_bOMA Then
    Dim Id As Integer
    Id = GetDlgItemID(RepMain.OMAMessage.hwnd)
    g_bResult = objRep.Initialize(1, RepMain.hwnd, Id, Network)
    If g_bResult = False Then
        MsgBox "Replication Initialization Failed"
        GoTo End_Sub
    End If
End If
```

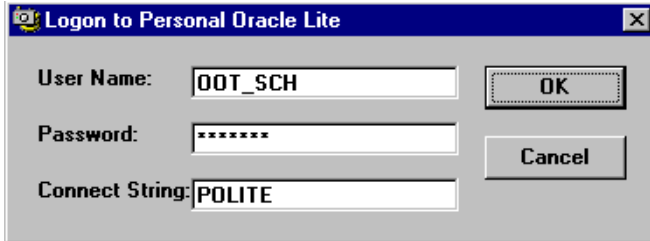
1.3. Connect to the master site

```
'Logon to database for SQL*Net connection. For OMA, we will be
'logging only when it is required
If g_bOMA = False Then
```

```
g_bMasterConnect = objRep.MasterConnectionOpen(szUserName, szPassword,
szHostName)
```

2. Connecting to the Snapshot Site

Use this form and the associated code to connect to the snapshot site (Personal Oracle Lite local database).



```
g_bStoreConnect = objRep.StoreConnectionOpen(szUserName, szPassword,
hostName)
```

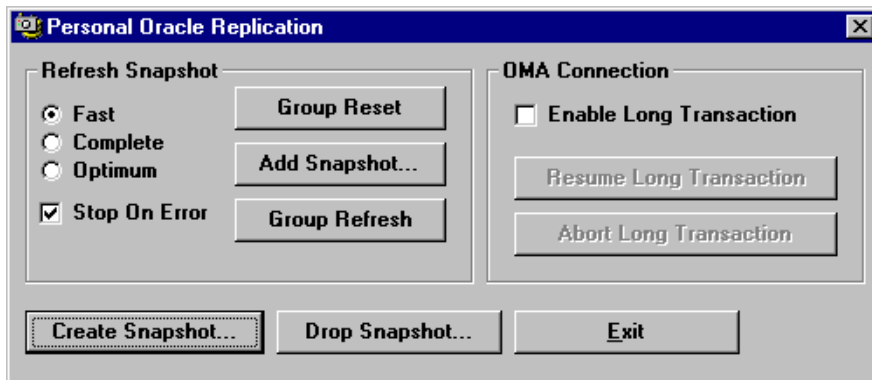
3. Performing Replication Operations

Use this form and the associated code to perform the following replication operations:

- 3.1. Initiate an instance of the Replication OLE control
- 3.2. Create a snapshot (calls Create Snapshot form)
- 3.3. Reset the snapshot refresh group
- 3.4. Add a snapshot (calls Add Snapshot form)
- 3.5. Refresh the snapshot refresh group
- 3.6. Get error information
- 3.7. Drop a snapshot (calls Drop Snapshot form)
- 3.8. Handle notification (if using Oracle Mobile Agents)
- 3.9. Enable the long transaction feature
- 3.10. Get long transaction status

3.11. Resume the pending operation

3.12. Abort the pending operation



3.1. Initiate an instance of the Replication OLE control

```
Set objRep = CreateObject("POLite.RepSvr")
```

3.2. Create a snapshot (calls Create Snapshot form, item 4.)

```
Private Sub CreateSnapshot_Click()  
    Load repCrt  
    repCrt.Show (1)  
End Sub
```

3.3. Reset the snapshot refresh group

```
Private Sub GroupReset_Click()  
    objRep.SnapshotGroupReset  
End Sub
```

3.4. Add a snapshot (calls Add Snapshot form, item 5.1)

```
Private Sub AddSnapshot_Click()  
    Load repAdd  
    repAdd.Caption = "Add Snapshot"  
    repAdd.Action.Caption = "Add"  
    repAdd.Show (1)  
End Sub
```

3.5. Refresh the snapshot refresh group

```
Private Sub GroupRefresh_Click()
```

```

Dim Tmp As Integer
If Fast = True Then
    Tmp = 0
End If
If Complete = True Then
    Tmp = 1
End If
If Optimum = True Then
    Tmp = 2
End If

'For OMA: Start the Transaction with MasterConnectionOpen
'For SQL*Net: we have already opened this connection, so skip it

If g_bOMA Then
    g_bResult = objRep.MasterConnectionOpen(g_szMasterUserName,
g_szMasterPassword, g_szMasterHostName)
End If

g_bResult = objRep.SnapshotGroupRefresh(Tmp, bStop)

'For OMA: MasterConnectionClose will trigger the OMA activities
'We don't need to do this for SQL*Net
If g_bOMA Then
    g_bResult = objRep.MasterConnectionClose
    EnableButtons False 'Disable the Buttons
End If

End Sub

```

3.6. Get error information

```

If Not g_bResult Then
    Dim errMsg As String
    errMsg = objRep.ErrorInfoGet
    MsgBox errMsg
End If

```

3.7. Drop a snapshot (calls Drop Snapshot form, item 5.2)

```

Private Sub DropSnapshot_Click()
    Load repAdd
    repAdd.Done.Caption = "Cancel"
    repAdd.Show (1)
End Sub

```

3.8. Handle notification (if using Oracle Mobile Agents)

```

Private Sub OMAMessage_Click()

```

```

Dim str As String
str = objRep.ErrorInfoGet
If str = "" Then
    MessageBox "OMA operation succeeded"
Else
    MessageBox str
End If
EnableButtons True
AbortLongTransaction.Enabled = False
ResumeLongTransaction.Enabled = False
End Sub

```

3.9. Enable the long transaction feature

```

Private Sub LongTransaction_Click()
    'Enable the long transaction. This call will set
    'a flag in the OMA Replication DLL
    objRep.LongTransactionEnable (LongTransaction)
End Sub

```

3.10. Get long transaction status

```

If g_bOMA = True Then
    OMARelated.Enabled = True
    LongTransaction.Enabled = True
    If objRep.ReplicationProgressGet = True Then
        LongTransaction.Value = 1
        ResumeLongTransaction.Enabled = True
        AbortLongTransaction.Enabled = True '
    End If
End If

```

3.11. Resume the pending operation

```

Private Sub ResumeLongTransaction_Click()
    If g_bOMA Then
        objRep.ReplicationResume
        EnableButtons False 'Disable the Buttons
        AbortLongTransaction.Enabled = False
        ResumeLongTransaction.Enabled = False
    End If
End Sub

```

3.12. Abort the pending operation

```

Private Sub AbortLongTransaction_Click()
    If g_bOMA Then
        objRep.ReplicationAbort
        ResumeLongTransaction.Enabled = False
    End If
End Sub

```

```

        AbortLongTransaction.Enabled = False
    End If

```

```

End Sub

```

4. Creating a Snapshot

Use this form and the associated code to create a snapshot.

```

'Specify whether simple or complex snapshot
Dim nType As Integer
If Simple = True Then
    nType = 0
Else
    nType = 1
End If

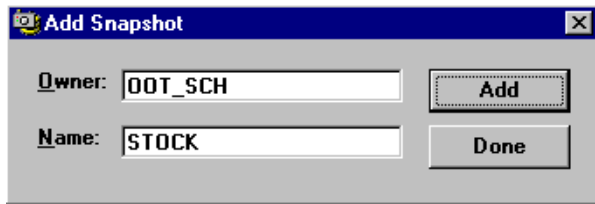
'Create the snapshot
g_bResult = objRep.SnapshotStoreCreate(m_ownerStore, m_nameStore,
m_ownerMaster, m_nameMaster, m_query, bUpdate, nType)

```

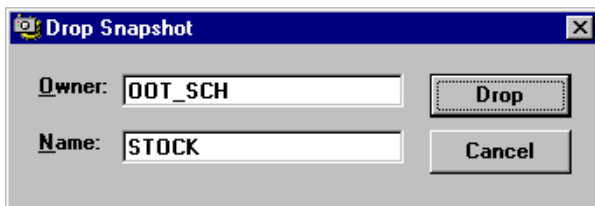
5. Adding/Dropping Snapshots

Use these forms and the associated code perform the following replication operations:

- 5.1. Add a snapshot to the refresh group
- 5.2. Drop a snapshot from the refresh group



The 'Add Snapshot' dialog box has a blue title bar with a yellow icon and a close button. It contains two text input fields: 'Owner' with the value 'OOT_SCH' and 'Name' with the value 'STOCK'. To the right of the 'Owner' field is an 'Add' button, and to the right of the 'Name' field is a 'Done' button.



The 'Drop Snapshot' dialog box has a blue title bar with a yellow icon and a close button. It contains two text input fields: 'Owner' with the value 'OOT_SCH' and 'Name' with the value 'STOCK'. To the right of the 'Owner' field is a 'Drop' button, and to the right of the 'Name' field is a 'Cancel' button.

5.1. Add a snapshot to the refresh group

```
objRep.SnapshotGroupAdd m_owner, m_name
```

5.2. Drop a snapshot

```
g_bResult = objRep.SnapshotStoreDrop(m_owner, m_name)
```


Index

A

- adding snapshots to refresh group, 22
- asynchronous replication, 16
- audience for this guide, iv

B

- base tables, pre-creating snapshot, 14
- bi-directional replication, 3

C

- complete refresh, 6
- complex snapshots, 2
- configuring
 - Oracle Mobile Agents, 17
 - SQL*Net, 16
- conflict resolution, 7
- conflicts
 - delete, 8
 - uniqueness, 8
 - update, 8
- connecting to databases, 20
- connectivity for replication, 15
- Continue-On-Error condition, 7

- creating snapshots, 21

D

- databases
 - connecting to, 20
 - master site, 2
 - Oracle, 2
 - Personal Oracle Lite, 2
 - snapshot site, 2
- delete conflicts, 8
- deploying replication applications, 25
- dropping snapshots, 22

E

- error conditions
 - Continue On Error, 7
 - handling, 23
 - Stop on Error, 7
 - types, 6
- ErrorInfoGet() method, 37

F

- fast path through this guide, vi
- fast refresh, 5
 - creating snapshot logs for, 13

H

- how to use this guide, v

I

- installing
 - Oracle Mobile Agents, 17
 - SQL*Net, 16

J

- join, 2, 35

L

- local database, 2
- logs, snapshot, 13
- logs, snapshot transaction, 4
- Long Transaction feature, 24

M

- master site
 - definition, 2
 - preparing for replication, 12
 - preparing master tables, 13
 - snapshot logs, 26
 - support for REPAPI, 12
 - verifying replication support, 12
- master tables
 - definition, 2
 - preparing for replication, 13
- MasterConnectionClose() method, 37

- MasterConnectionOpen() method, 34
- methods, Replication OLE Control, 33
- MLOG\$_tablename, 13, 26

N

- Navigator, 20
- notational conventions in this guide, vi

O

- optimum refresh, 6
- Oracle Mobile Agents
 - configuring, 17
 - handling notification and, 24
 - installing, 17
 - Long Transaction feature and, 24
 - replication, 16
 - setting up, 29
 - working with, 24

P

- pre-creating snapshot base tables, 14

R

- read-only snapshots
 - compared to updatable, 3
 - definition, 2
- recreating snapshots, 26
- refresh
 - complete, 6
 - errors, 6
 - fast, 5
 - group, 6
 - optimum, 6
 - scheduling, 23
- refresh group, 6
- refreshing snapshots, 22

- Refresh-In-Progress mode, 7
- remote database, 2
- REPAPI, 12
- replication
 - asynchronous, 16
 - bi-directional, 3
 - conflict resolution, 7
 - connectivity, 15
 - definition, 2
 - engine, 12
 - errors, 6
 - handling notification and, 24
 - Long Transaction feature, 24
 - more information on, vii
 - overview, 2
 - performing programmatically, 20
 - performing with the Navigator, 20
 - preparing master site, 12
 - preparing master tables, 13
 - preparing snapshot site, 13
 - synchronous, 15
 - uni-directional, 3
 - with Oracle Mobile Agents, 16
 - with SQL*Net, 15
- replication applications
 - deploying, 25
 - sample, 27
- replication engine (REPAPI), 12
- Replication OLE Control (OCX)
 - description, 8
 - methods, 33
- replicaton support, verifying, 12

S

- sample replication application
 - commented, 41
 - running, 27
- simple snapshots, 2
- SLOG, 26
- snapshot logs
 - master site, 26

- snapshot logs, creating, 13
- snapshot refresh
 - complete, 6
 - definition, 5
 - description, 5
 - errors, 6
 - fast, 5
 - group, 6
 - optimum, 6
 - scheduling, 23
 - types, 5
- snapshot refresh group
 - adding snapshots, 22
- snapshot site
 - definition, 2
 - preparing for replication, 13
 - snapshot transaction logs, 4
- snapshot transaction logs, 4
 - clearing, 26
- SnapshotGroupAdd() method, 36
- SnapshotGroupRefresh() method, 36
- SnapshotGroupReset() method, 36
- snapshots
 - adding to snapshot refresh group, 22
 - complete refresh, 6
 - complex, 2
 - creating, 21
 - definition, 2
 - dropping, 22
 - fast refresh, 5
 - optimum refresh, 6
 - pre-creating base tables, 14
 - read-only, 2–3
 - recreating, 26
 - refresh types, 5
 - refreshing, 5, 22
 - simple, 2
 - transaction logs, 4
 - types, 2
 - updatable, 2–3
 - working with, 21
- SnapshotStoreCreate() method, 35

- SnapshotStoreDrop() method, 36
- SQL*Net
 - configuring, 16
 - installing, 16
 - replication, 15
- Stop-On-Error condition, 7
- StoreConnectionClose() method, 38
- StoreConnectionOpen() method, 34
- synchronous replication, 15

T

- transaction logs, snapshot, 4

U

- uni-directional replication, 3
- uniqueness conflicts, 8
- updatable snapshots
 - compared to read-only, 3
 - definition, 2
 - restrictions on, 5
- update conflicts, 8
- USLOG, 4, 26
- USLOG\$_snapshotname, 4, 26

Reader's Comment Form

Name of Document: Replication with Personal Oracle Lite™ A53900-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Do you better access to the information that you need? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have suggestions for improvement, please indicate the topic, chapter, and page number below:

Please send your comments to:

Personal Oracle Lite Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065

or e-mail comments to: mgaydos@nc.com

If you would like a reply, please give your name, address, and telephone number below:

Thank you for helping us improve our documentation.