Oracle® Database Lite

Tools and Utilities Guide

10*g* (10.0.0)

Part No. B12263-01

June 2004

ORACLE®

Oracle Database Lite Tools and Utilities Guide 10*g* (10.0.0)

Part No. B12263-01

Primary Author: Mujeeb Ahmed

Contents

Send Us Your Comments vii

Preface ix

Intended Audience ix
Documentation Accessibility ix
Structure ix

1 Introduction

1.1 Overview 1-1
1.2 Packaging Wizard 1-1
1.3 Database Tools and Utilities 1-1
1.4 Consperf 1-2

2 Using the Packaging Wizard

2.1 Using the Packaging Wizard for Web Applications 2-1
2.1.1 Starting the Packaging Wizard 2-2
2.1.2 Specifying New Application Details 2-3
2.1.3 Listing Application Files 2-5
2.1.3.1 Compile JSP (For Web Applications Only) 2-6
2.1.3.2 Filters 2-7
2.1.4 Adding Servlets (For Web Applications Only) 2-7
2.1.5 Entering Database Information 2-8
2.1.6 Defining Application Roles 2-9
2.1.7 Defining Snapshots for Replication 2-10
2.1.7.1 Creating New Snapshots 2-12
2.1.7.2 Creating Indexes for Snapshots 2-13
2.1.7.3 Importing Snapshots 2-14
2.1.7.4 Editing Snapshots 2-15
2.1.8 Defining Sequences for Replication 2-16
2.1.8.1 Importing Sequences 2-18
2.1.9 Defining Application DDLs 2-20
2.1.9.1 Importing Views and Index Definitions 2-21
2.1.10 Editing Applications 2-21
2.1.11 Troubleshooting 2-22
2.2 Packaging Wizard Synchronization Support 2-22

3 Database Tools and Utilities For Win32 and WinCE Platforms

3.1 MQSL ... 3-2
3.1.1 MSQL for Windows 32 ... 3-2
3.1.1.1 Populating your Database Using Mobile SQL ... 3-2
3.1.1.2 Mobile SQL ... 3-2
3.1.1.3 Database Access ... 3-3
3.1.1.4 Starting Mobile SQL ... 3-3
3.1.1.5 SET TERM {ON | OFF} ... 3-3
3.1.1.6 SET TIMING {ON | OFF} ... 3-3
3.1.1.7 SET VERIFY {ON | OFF} ... 3-3
3.1.2 MSQL for Windows CE ... 3-3
3.1.2.1 Using Mobile SQL ... 3-3
3.1.2.2 Database Access ... 3-4
3.1.2.3 Starting Mobile SQL ... 3-4
3.2 Database Tools and Utilities for PALM ... 3-4
3.2.1 Overview ... 3-4
3.2.2 Installing ValidateDB ... 3-4
3.2.3 Running ValidateDB ... 3-4
3.2.4 Sending Corrupted Databases ... 3-5
3.2.5 Must Send Databases ... 3-5
3.3 CREATEDB ... 3-6
3.4 DECRYPDB ... 3-7
3.5 ENCRYPDB ... 3-8
3.5.1 Synchronizing with an Encrypted Database ... 3-10
3.6 MIGRATE ... 3-10
3.7 ODBC Administrator and the Oracle Database Lite ODBC Driver ... 3-11
3.7.1 Adding a DSN Using the ODBC Administrator ... 3-13
3.7.2 Adding a DSN which points to Read-Only Media (CD-ROM) ... 3-13
3.8 ODBINFO ... 3-13
3.9 OLLOAD ... 3-15
3.10 REMOVEDB ... 3-18
3.11 VALIDATEDB ... 3-18

4 Load Utility

4.1 Overview ... 4-1
4.2 Syntax ... 4-1
4.3 Keywords and Parameters ... 4-1
4.3.1 Options ... 4-2
4.3.1.1 Comments ... 4-3

5 Consolidator Performance (Consperf) Utility

5.1 Overview ... 5-1
5.2 Generating Timing Statistics for Publications ... 5-2
5.3 Generating Explain Plans for Publications ... 5-3
5.4 Automatically Tuning Publication Properties ... 5-3
5.5 Analyzing Mobile Server Objects for Cost Based Optimizer ... 5-4

5.6 Usage Model 5-4
5.7 Consperf Parameters 5-5

Index

Send Us Your Comments

Oracle Database Lite Tools and Utilities Guide 10*g* (10.0.0)

Part No. B12263-01

Oracle 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?
- 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 any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: helplite_ca@oracle.com
- FAX: (650) 506-7355. Attn: Oracle Database Lite
- Postal service:

  Oracle Corporation
  Oracle Database Lite Documentation
  500 Oracle Parkway, Mailstop 1op2
  Redwood Shores, CA 94065
  U.S.A.

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This preface introduces you to the *Oracle Database Lite Tools and Utilities Guide*, discussing the intended audience, documentation accessibility, and structure in this document.

Intended Audience

This manual is intended for application developers as the primary audience and for database administrators who are interested in application development as the secondary audience.

Documentation Accessibility

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

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

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

Structure

This guide includes the following topics.

- Chapter 1, "Introduction"

  Provides an introduction to all the Tools and Utilities that are used during application development.

- Chapter 2, "Using the Packaging Wizard"

  Enables you to package and publish your mobile applications using the Packaging Wizard.

- Chapter 3, "Database Tools and Utilities For Win32 and WinCE Platforms"

  Describes how to use the database utilities for the Windows 32 and Windows CE platforms.

- Chapter 4, "Load Utility"

  Describes the Oracle Database Lite Load utility.

- Chapter 5, "Consolidator Performance (Consperf) Utility"

  Provides a reference to the Consolidator Performance (Consperf) utility which is used to profile Consolidator publications.

1 Introduction

This document provides an introduction to all the tools and utilities that are used during application development. Topics include:

- Section 1.1, "Overview"
- Section 1.2, "Packaging Wizard"
- Section 1.3, "Database Tools and Utilities"

1.1 Overview

Oracle Database Lite is shipped with a set of tools and utilities that enable tasks relevant to developing, packaging, publishing and synchronizing mobile applications on the Mobile Server. The following sections provide a brief introduction to these tools and utilities.

1.2 Packaging Wizard

The Packaging Wizard is a graphical tool that enables you to perform the following tasks.

- Create a new mobile application.
- Edit an existing mobile application.
- Publish an application to the Mobile Server.

When you create a new mobile application, you must define its components and files. In some cases, you may want to edit the definition of an existing mobile application's components. For example, if you develop a new version of your application, you can use the Packaging Wizard to update your application definition. The Packaging Wizard also enables you to package application components in a .jar file which can be published using the Mobile Manager. The Packaging Wizard also enables you to create SQL scripts which can be run to create base tables in the Oracle database.

1.3 Database Tools and Utilities

This section describes how to use the following database tools and utilities.

- Allows databases to be created with linguistic sort capability enabled
- Create Oracle Database Lite databases
- Decrypt Oracle Database Lite
- Remove Java classes from Oracle Database Lite

- Encrypt Oracle Database Lite
- Load Java classes into Oracle Database Lite
- Migrate to Oracle Database Lite from a previous release
- Connect to Oracle Database Lite databases
- Manage ODBC connections
- Identify version number and volume ID of Oracle Database Lite
- Load data from an external file into a table in Oracle Database Lite, or unload (dump) data from a table in Oracle Database Lite to an external file.
- Remove Oracle Database Lite
- Validate the structure of Oracle Database Lite

1.4 Consperf

The Consperf utility is used to profile Consolidator publications. Application developers and administrators can use this utility to analyze performance of publications and identify potential bottlenecks during publication. This tool enables users to perform four primary functions:

- Generate Timing Statistics for Publications
- Generate Explain Plans for Publications
- Automatically Tune Publication Properties
- Analyze Mobile Server Objects for Cost Based Optimizer

During the Synchronization and MGP process, the Consolidator wraps publication item queries in templates to determine incremental changes. With complex snapshot queries, these templates can confuse the Oracle optimizer frequently and may result in poor execution plans. The Consperf utility exposes such templates and profiles their performance in conjunction with actual publication item queries. The Consperf utility generates SQL explain plans for each query.

2

Using the Packaging Wizard

This chapter enables you to package and publish your mobile applications using the Packaging Wizard. Topics include:

- Section 2.1, "Using the Packaging Wizard for Web Applications"
- Section 2.2, "Packaging Wizard Synchronization Support"

2.1 Using the Packaging Wizard for Web Applications

The Packaging Wizard is a graphical tool that enables you to perform the following tasks.

- Create and publish a new mobile application.
- Edit an existing mobile application.
- Package a mobile application for easy deployment.
- Publish a mobile application to the Mobile Server Repository.

When you create a new mobile application, you define its components and publish them to the Mobile Server Repository. In some cases, you may want to edit the definition of an existing mobile application's components. For example, if you develop a new servlet for your application, you can use the Packaging Wizard to add the servlet to your application definition and then publish the modified application to the Mobile Server Repository. The Packaging Wizard also enables you to package application components in a `.jar` file for easy deployment.

Additionally, the Packaging Wizard supports the Web Application Archive (WAR) file format including the `web.xml` descriptor file. WAR files can be created using tools such as *J*Developer, according to the Java Servlet Specification 2.3.

The following sections describe how to use the Packaging Wizard. Topics include:

- Section 2.1.1, "Starting the Packaging Wizard"
- Section 2.1.2, "Specifying New Application Details"
- Section 2.1.3, "Listing Application Files"
- Section 2.1.4, "Adding Servlets (For Web Applications Only)"
- Section 2.1.5, "Entering Database Information"
- Section 2.1.6, "Defining Application Roles"
- Section 2.1.7, "Defining Snapshots for Replication"
- Section 2.1.8, "Defining Sequences for Replication"

- Section 2.1.9, "Defining Application DDLs"
- Section 2.1.10, "Editing Applications"
- Section 2.1.11, "Troubleshooting"

2.1.1 Starting the Packaging Wizard

To launch the Packaging Wizard, enter the following using a Command Prompt window.

```
wtgpack
```

As Figure 2–1 displays, the Packaging Wizard appears and displays the Make a Selection dialog. Using this dialog, you can create, edit, open, or delete a packaged application using the following features.

Figure 2–1 Packaging Wizard - Make A Selection Dialog

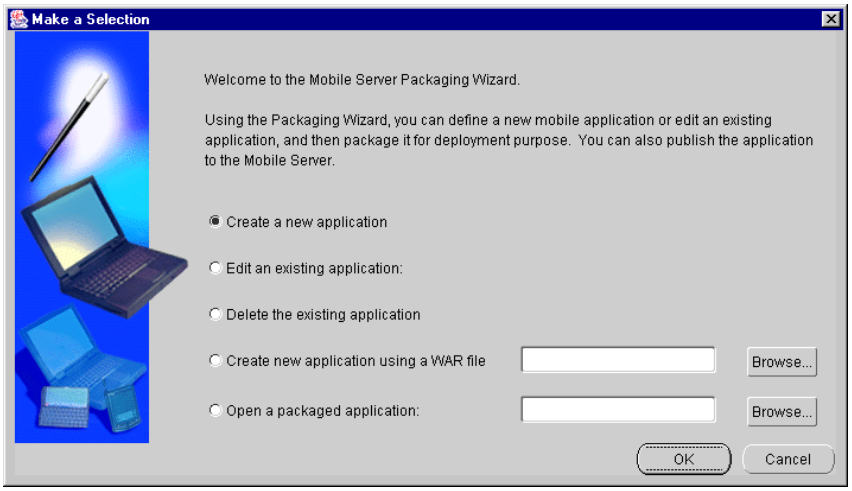


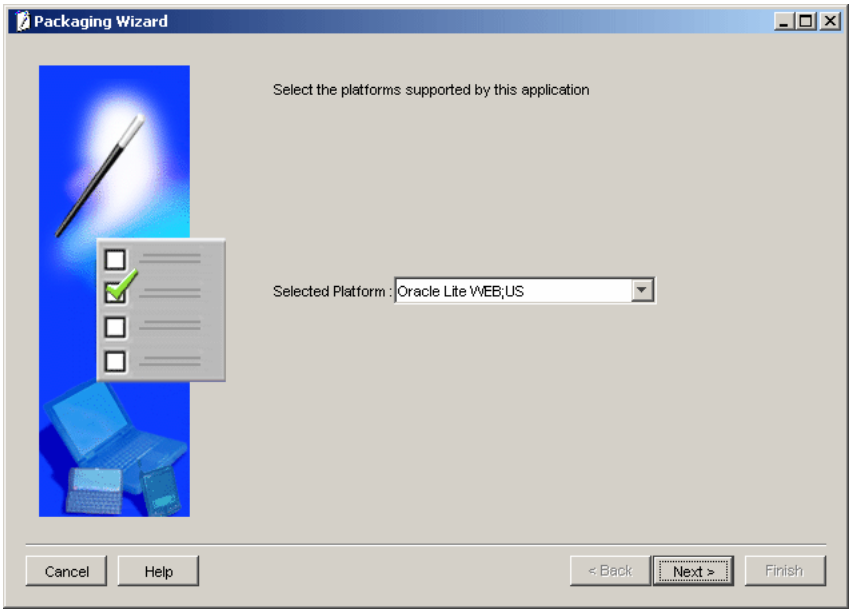
Table 2–1 describes the Make a Selection dialog.

Table 2–1 Make a Selection Dialog

Feature	Description
Create a new application	If selected, this option enables you to define a new application.
Edit an existing application	If selected, this option enables you to edit an existing application. Users can select an existing application from the list displayed.
Delete an existing application	If selected, this option enables you to delete an existing application from the XML file, but not from the Mobile Server Repository.
Creating a new application using a WAR file	If selected, this option enables you to create an application using a Web Application Archive (WAR) file. You can enter the name of the WAR file or locate it using the 'Browse' button.
Open a Packaged application	If selected, this option enables you to select an application that has been packaged a JAR file. You can enter the name of the packaged application or locate it using the 'Browse' button.

Using the 'Select a Platform' dialog, select the platform for which you want to package your application. As Figure 2–2 displays, this dialog enables you to specify a platform. If you are packaging a WAR file, this dialog only displays web based platforms.

Figure 2–2 Select a Platform Dialog



2.1.2 Specifying New Application Details

Using the Application dialog, you can name a new Web-to-Go application and specify its storage location on the Mobile Server. As Figure 2–3 displays, the Application dialog includes the following fields.

Figure 2–3 Application Dialog

Table 2–2 describes the Application dialog.

Table 2–2 Application Dialog Description

Field Name	Description	Required
Application Name	The name of the new mobile application. When packaging a WAR file, the application name must be set to the value of the element `<display-name>`, which can be found under the main element `<web-app>` in the file `web.xml`.	Yes
Virtual Path	A path that is mapped from the root directory of the server repository to the mobile application itself. The virtual path eliminates the need to refer to the application's entire directory structure. It indicates that all of the subdirectories and all of the files that are in the virtual path will be uploaded exactly as they are in the directory structure to the Mobile Server Repository when the application is published. It also provides the application with a unique identity. **Application Root Directory** As Figure 2–3 displays, the name `/tutorial` indicates the virtual path of the application. The name that you enter as the virtual path of the application becomes the **application root directory** within the Mobile Server Repository, when the application is published. Consequently, you can specify the application root directory by the name that you enter in the virtual path field. This name can be different from the application name, but should not contain spaces. For example, your application name can be 'Sales Office' and your virtual path '/Admin'. In this case, '/Admin' becomes the name of the application root directory within the Mobile Server Repository. The application root directory is the location where the actual application files are stored within the Mobile Server Repository. When the administrator publishes the application, the Packaging Wizard automatically uses the name that you entered in the virtual path as the name of the application root directory in the Mobile Server Repository. However, the administrator can change the name of the application root directory in the Mobile Server Repository by entering a different name for it when the administrator publishes the application.	Yes
Description	A brief description of the mobile application. When packaging a WAR file, the description must be set to the value of the element `<description>` found under the main element `<web-app>` in the `web.xml` file.	Yes

Table 2–2 (Cont.) Application Dialog Description

Field Name	Description	Required
Application Classpath [Web Applications Only]	The application classpath specifies where the classes (servlets, beans) for the application are located. The default application classpath is always the application root directory. To specify additional locations that the Mobile Server can search for application classes, add other directories or jar and zip files to the application classpath for web applications. Entries must be separated by semicolons (;) In addition, Web-to-Go automatically appends the following to the application classpath: 1. Application root directory 2. Classpath as specified in the 'Application' dialog in the Packaging Wizard 3. Classes located under `WEB-INF/classes` 4. All jar and zip files located in the directory `WEB-INF/lib` 5. Classes located under the directory `/shared/WEB-INF/classes` 6. All jar and zip files located in the directory `/shared/WEB-INF/lib` 7. `SYSTEM` classpath	No
Default Page [Web Applications Only]	The server location of the Web page that functions as the mobile application's entry point. This is a relative path to the repository directory. For example, if the server directory is `/apps` and the default page is `index.htm`, the Default Page is `/apps/index.htm`. The default page can be a servlet. A generic page is issued if the user does not specify a default page. When packaging a WAR file, the default page must be set to the value of the element `<welcome-file-list>` in the `web.xml` file.	Yes
Local Application Directory	The directory on the local machine that contains all components of the application. You can type this location or locate it using the 'Browse' button. During development, the application root directory is set to the local application directory.	Yes
Icon [Web Applications Only]	The GIF image of the mobile application is used as the application icon in the mobile workspace. Users may enter the icon name in the corresponding field or locate it using the 'Browse' button. When packaging a WAR file, the description field must be set to the value of the element `<large-icon>` as a primary choice or `<small-icon>` as a secondary choice found under the main element `<web-app>` in the `web.xml` file.	

2.1.3 Listing Application Files

Use the Files panel to list your application files and to specify their location on the local machine. The Packaging Wizard analyzes the contents of the Local Application Directory and displays each file's local path. As Table 2–3 describes, the Files tab contains the following field.

Figure 2–4 displays the Files tab.

Figure 2–4 Files Tab

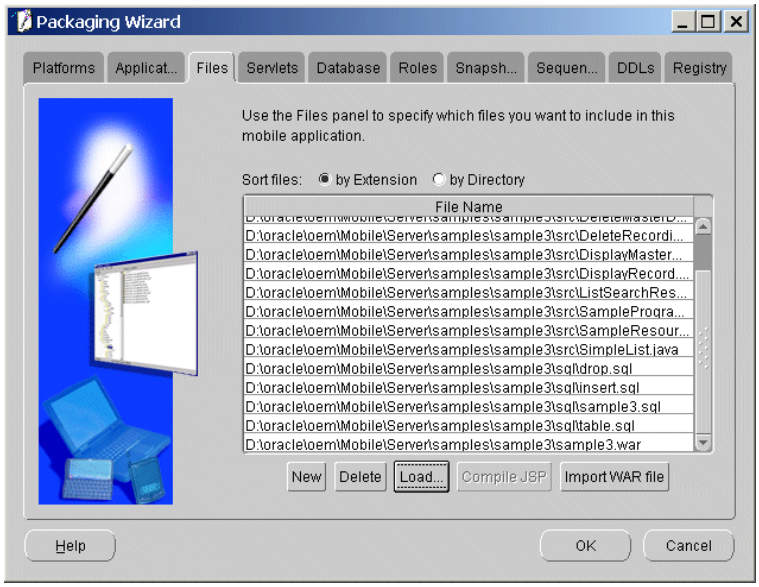


Table 2–3 Files Tab Description

Field	Description	Required
Local Path	The absolute path of each mobile application file. Each entry on the list includes the complete path of the individual file or directory.	Yes

You can add, remove, load, or compile any of the files that are listed in the 'Files' dialog. If you are creating a new application, the Packaging Wizard automatically analyzes and loads all files listed under the local directory when you proceed to the 'Files' dialog. If you are editing an existing application, upload your individual application files using the 'Load' button.

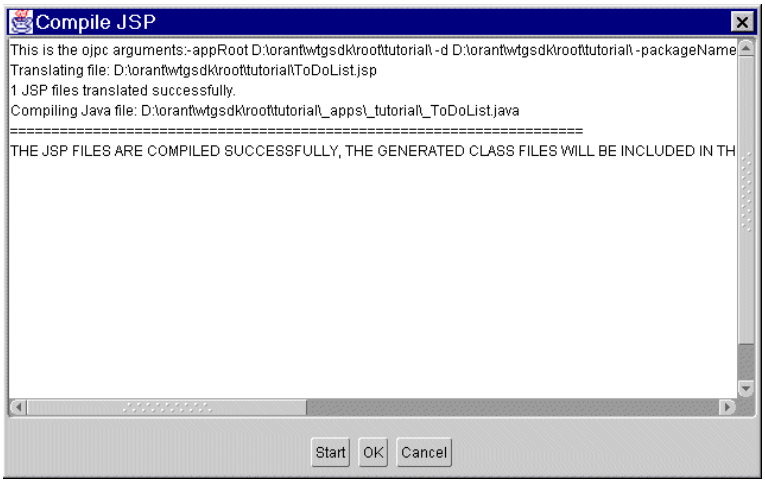
If you are importing a WAR file into an existing application, click the **Import WAR File** button on the 'Files' tab. Once you have specified the location of the WAR file, the 'Files' tab displays content of the WAR file.

2.1.3.1 Compile JSP (For Web Applications Only)

The 'Compile JSP' button enables you to compile your JSP files for deployment. If you click the 'Compile JSP' button, the following 'Compile JSP' dialog appears with detailed compilation information. If there are any errors, you should correct the JSP files before proceeding.

Figure 2–5 displays the Compile JSP Dialog.

Figure 2–5 Compile JSP Dialog



You can sort the files by their extensions or by the directory in which they are located. To sort files, click the 'By Extension' or 'By Directory' options.

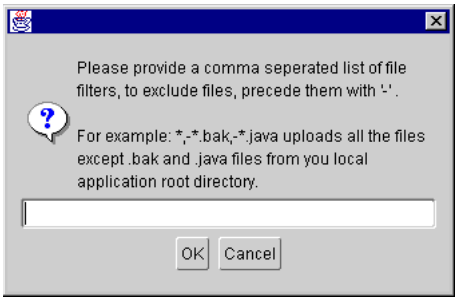
2.1.3.2 Filters

When you click the 'Load' button, the 'Input' dialog appears. You can use the 'Input' dialog to create a comma-separated list of filters that either include or exclude application files from the upload process. To exclude a file, type a preceding minus sign (-) before the file name. For example, to load all files but exclude files with the `.bak` and `.java` suffixes, enter the following.

```
*,-*.bak,-*.java
```

Figure 2–6 displays the Input dialog.

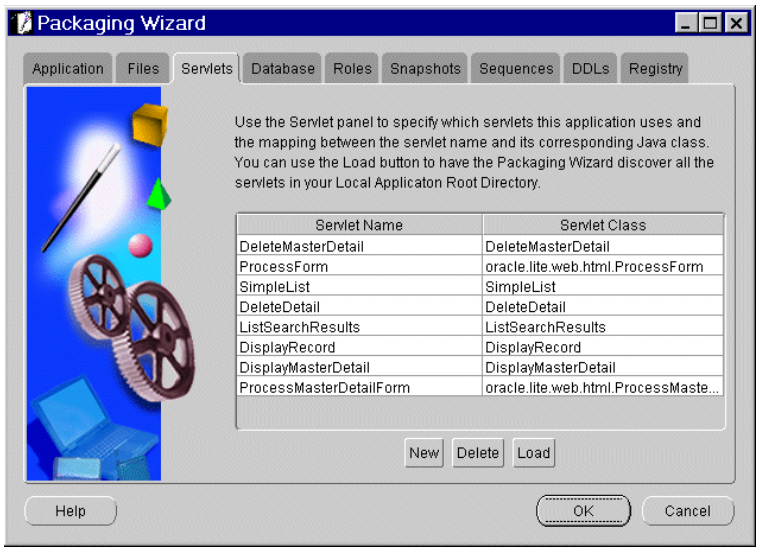
Figure 2–6 Input Dialog



2.1.4 Adding Servlets (For Web Applications Only)

The Packaging Wizard analyzes servlets in the File tab and defines them on the Mobile Server. As displayed in Figure 2–7, you can view your application's servlets in the Servlets tab.

Figure 2–7 Servlets Tab



As described in Table 2–4, the 'Servlets' tab includes the following fields.

Table 2–4 Servlets Tab Description

Field	Description	Required
Servlet Name	The servlet's name. For example: `DeleteDetail`. You will then refer the servlet as: `application_virtualpath/servlet name`	Yes
Servlet Class	The fully qualified class of the servlets to be added.	Yes

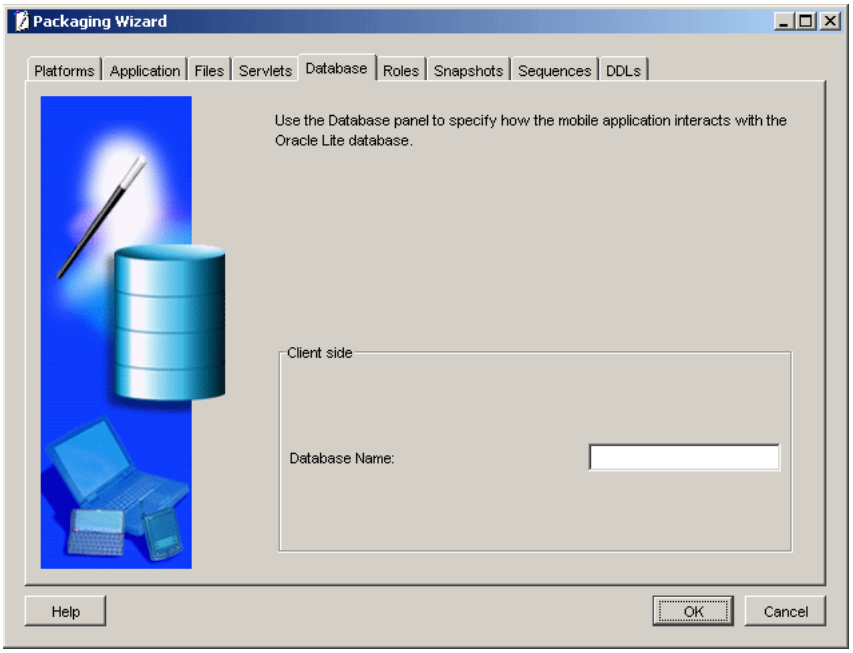
Using the 'Servlets' tab, you can add, remove, or load any servlets that are listed under the 'Servlets' tab. If you are creating a new application, the Packaging Wizard automatically lists all 'Servlets' based on files that are listed in the 'Files' tab. If you are editing an existing application, use the 'Load' button to locate and load individual servlets.

2.1.5 Entering Database Information

Using the Database tab, you can provide connection information and specify how the mobile application user connects to the replication master groups on the Oracle server.

Table 2–8 displays the Database tab.

Figure 2–8 Database Tab



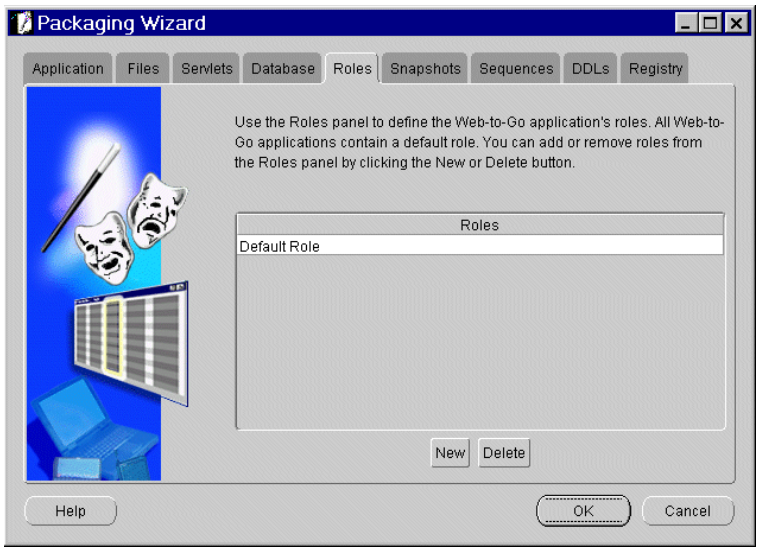
Enter the database name that you want to create on the client side. For example, a native Windows 32 application accesses the client database with this name. However, this is not required for web applications.

2.1.6 Defining Application Roles

Use the 'Roles' tab to define the Mobile Server application's roles. Developers create roles in the application's code and the Packaging Wizard re-declares them for the Oracle database. After you publish the application to the Mobile Server, you can assign roles to users and groups, using the Mobile Manager.

Figure 2–9 displays the Roles tab.

Figure 2–9 Roles Tab



As described in Table 2–5, the Roles tab includes the following field.

Table 2–5 Roles Tab Description

Field	Description
Roles	Assigns roles to the Web-to-Go/Mobile Server application.

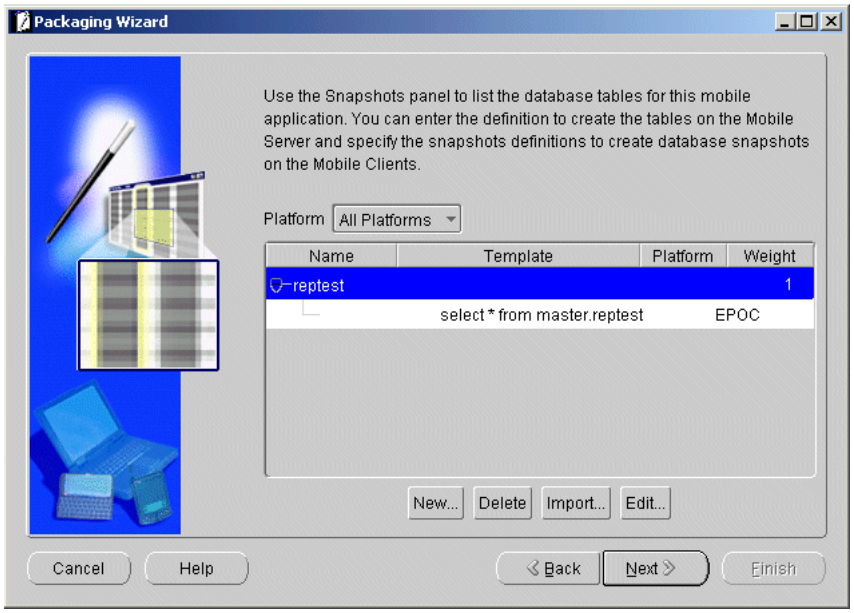
All Web-to-Go/Mobile Server applications contain a default role. You can add or remove roles from the Roles dialog using the 'New' or 'Delete' button.

2.1.7 Defining Snapshots for Replication

Use the Snapshots tab to create replication snapshots for your application. A snapshot must have the same name as the database object such as a table or view. It must be unique across all applications. However, you must ensure that you use unique names when creating database objects. The Packaging Wizard enables you to create snapshots for the chosen platform. When you specify a view as the base object type, the Packaging Wizard enables you to specify the Parent Hint, Virtual Primary Hint, and the Primary Key Hint. For Web-to-Go, use the Windows 32 platform.

Figure 2–10 displays the Snapshots tab.

Figure 2–10 Snapshots Tab



Note: Once you have specified a database connection, it is used for the remainder of your Packaging Wizard session. If you need to switch between an Oracle database and Oracle Database Lite, but have already established a connection, you must quit the Packaging Wizard application completely and run `wtgpack.exe` again.

Table 2–6 describes the Snapshots tab.

Table 2–6 Snapshots Tab Description

Field	Description	Required
Name	The name(s) of the snapshot(s) associated with the Web-to-Go/Mobile Server application. It must be the same name as the underlining database object.	Yes
Template	Lists available snapshot templates. The template is a SQL statement that is used to create the snapshot. The template may contain variables. After you publish the template to the Mobile Server, you can specify user-specific template variables using the Mobile Manager. However, you cannot modify snapshots in the Mobile Manager.	Yes
Weight	This is the order of tables to be replicated. For tables with a master-detail relationship, the master table needs to be replicated first and therefore should have a lower weight.	No

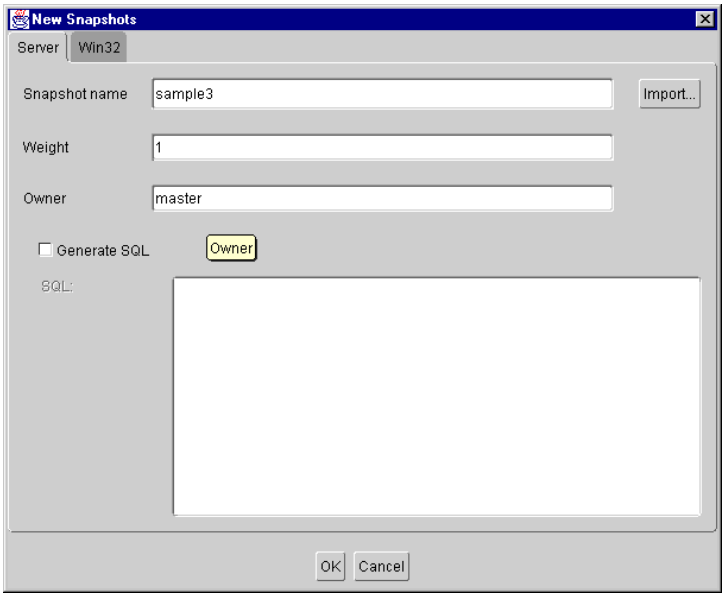
You can add or remove snapshots from the Snapshots tab using the 'New' or 'Delete' button. You can also import or edit snapshots using the 'Import' or 'Edit' button.

Note: You can import multiple snapshots from the Snapshots tab or import one when you create a new table from the 'New Table Dialog'.

2.1.7.1 Creating New Snapshots

To create new snapshots, click 'New'. The 'New Snapshots' dialog appears. As Figure 2–11 displays, if you click the Server tab, the Server dialog appears, which contains fields for snapshot name, weight, owner, and SQL, as well as a check box for generating SQL.

Figure 2–11 New Snapshots Dialog - Server Tab



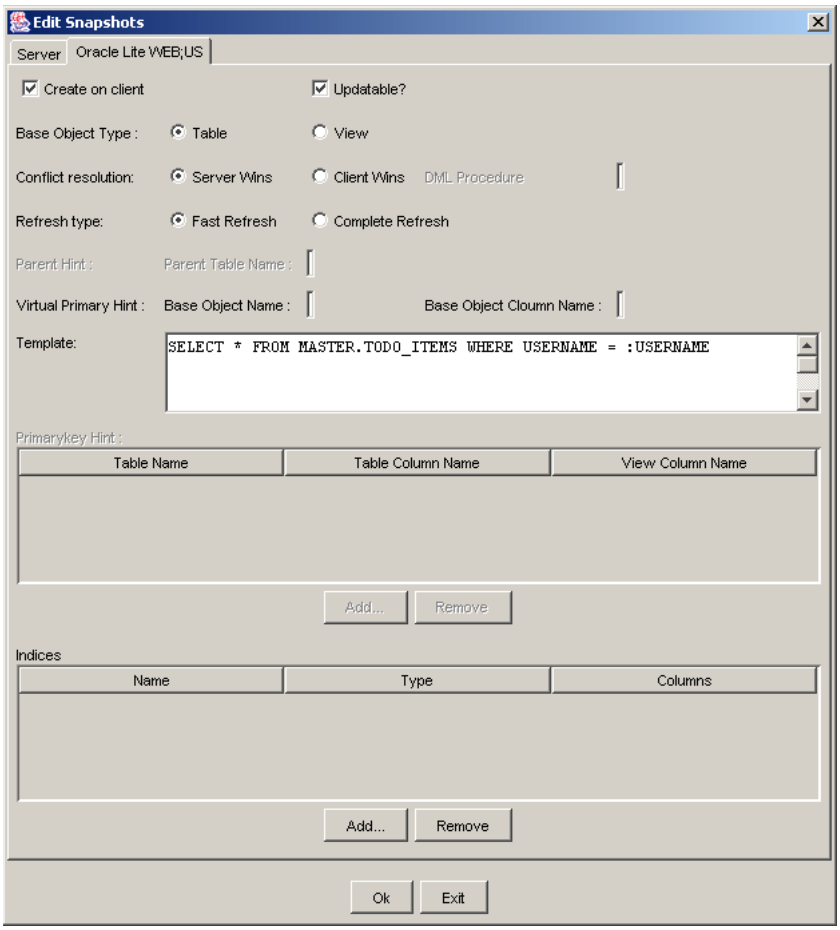
For a description of Weight, see Section 2.1.7, "Defining Snapshots for Replication".

If you enable Generate SQL, you should provide a `CreateTable` SQL statement to construct the database table. Enter the SQL statement in the SQL field. At the end of the defining phase, you will have the option to generate a SQL file.

Use the Win32 tab for the Mobile Client for Web-to-Go.

If you click the Win32 tab, the following dialog appears.

Figure 2–12 Edit Snapshots Dialog - Win32 Tab



Create a new snapshot on the Mobile Client for Web-to-Go by modifying the following features in the New Snapshots dialog.

As Figure 2–7 describes, the New Snapshots dialog displays the following information.

Table 2–7 New Snapshots Dialog Description

Field	Description
Updatable	When selected, this check box creates an updatable snapshot of the named table.
Template	Displays the snapshot template for the named table. You can modify the snapshot template. Administrators can instantiate variables for different users to this template using the Mobile Manager. For more information about template variables, see Section 2.1.7, "Defining Snapshots for Replication".

2.1.7.2 Creating Indexes for Snapshots

To create an index for a snapshot using the Packaging Wizard, use the following procedure.

1. From the Snapshots dialog, select the Edit button to create an index from an existing snapshot, or the New button for creating a new snapshot and new index.

2. Select the platform tab on the dialog which appears, for example Win 32. The SQL statement which defines your snapshot appears in the 'Template' field. Below that is an 'Indices' table; to create a new index, select the 'New' button beneath this table.

   As Table 2–8 describes, enter values in the Win32 tab of the Edit Snapshots dialog.

Table 2–8 Win32 Tab - Edit Snapshots Dialog

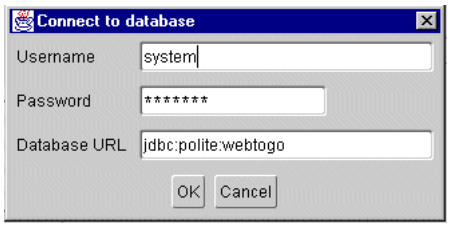
Field	Description
Create on Client	If selected, creates the snapshot on the client machine.
Updatable	If selected, creates an updatable snapshot of the specified table or view.
Base Object Type	Select **Table** to include a table as the base object type. or Select **View** to include a view as the base object type.
Conflict Resolution	Select **Server Wins** to specify conflict resolution in favour of the server. or Select **Client Wins** to specify conflict resolution in favour of the client.
DML Procedure	To specify the DML procedure, enter the name of the Callout Package for DML operation.
Refresh Type	Select **Fast Refresh** to specify a quick refresh of the snapshot. or Select **Complete Refresh** to specify a complete refresh of the snapshot.
Parent Hint	To specify the parent hint, enter the **Parent Table Name**.
Virtual Primary Hint	To specify the virtual primary hint, enter the **Base Object Name** and **Base Object Column** in the corresponding fields.
Template	Displays the snapshot template for the named table. You can modify the snapshot template. Administrators can instantiate variables for different users to this template using the Mobile Manager. For more information about template variables, see Section 2.1.7, "Defining Snapshots for Replication".
Primary Key Hint	This section displays the **table name**, **column name**, and **mapping column name** of the snapshot.
Indices	This section displays the **name**, **type**, and **column name** of indices used in a snapshot.

3. There are three columns in the 'Indices' table:
   a. Name - This is the name of the index.
   b. Type - Indexes can be Regular, Primary, or Unique. There is a drop down menu to select this.
   c. Columns - Enter the column name which the index uses.

2.1.7.3 Importing Snapshots

To import snapshots from an Oracle database or from Oracle Database Lite, click the 'Import' button. As Figure 2–13 describes, the database connection window appears if you have not specified a connection.

Figure 2–13 Connect to Database Dialog

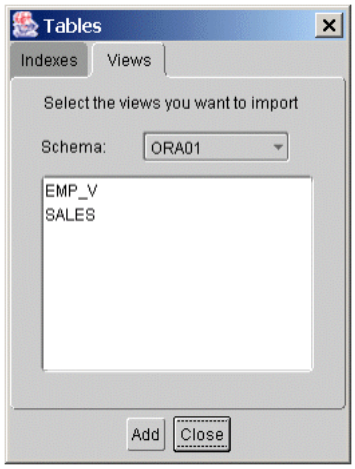


Enter the user name, password, and database URL for the Oracle database, or Oracle Database Lite from which you are importing your snapshot(s). The Tables window appears.

Note: Use the following format when entering the database URL for an Oracle database: `jdbc:oracle:oci8:@webtogo.world`. For Oracle Database Lite, use `jdbc:polite:webtogo`.

Figure 2–14 displays the Tables dialog.

Figure 2–14 Tables Dialog

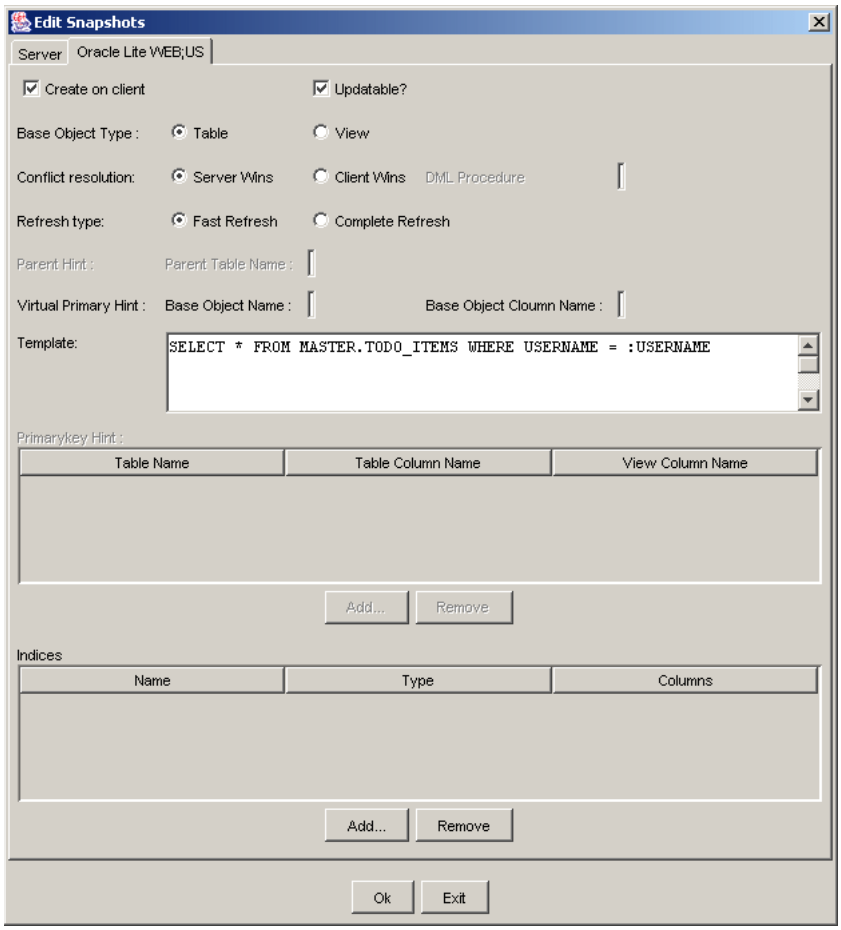


Click the Schema list and choose the required schema from the list displayed. The Tables dialog displays views associated with the chosen schema. Select the view that you need to import. Click Add and click Close.

2.1.7.4 Editing Snapshots

To edit a snapshot, select the snapshot from the Snapshots dialog and click Edit. As displayed in Figure 2–15, the Edit Snapshots dialog appears.

Figure 2–15 Edit Snapshots Dialog - Win32 Tab



As described in Table 2–9, edit the snapshot by modifying the following features of the Edit Table window:

Table 2–9 Edit Snapshots Dialog - Win32 Tab Description

Feature	Description
Create on Client	When selected, the checkbox allows you to edit the snapshot on the Mobile Client for Web-to-Go.
Updatable	When selected, this check box creates an updatable snapshot of the named table.
Template	Displays the snapshot template for the named table. You can modify the snapshot template. Administrators can instantiate variables for different users to this template using the Mobile Manager.

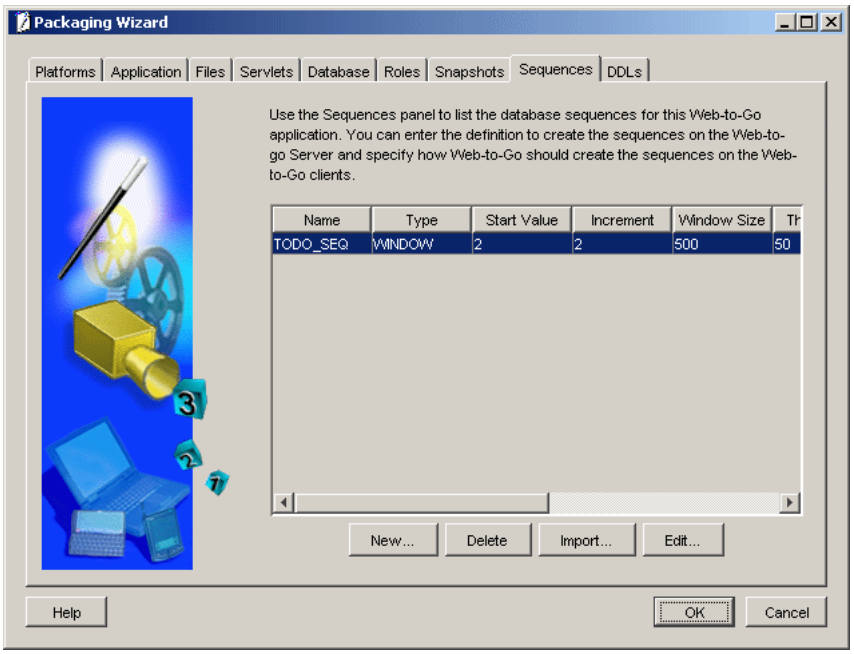
2.1.8 Defining Sequences for Replication

Use the Sequences dialog to define offline sequence support for the Web-to-Go application. Web-to-Go uses sequences to assign unique primary key values to an application before it disconnects and is in offline mode. These unique primary key values are used for replication when the client goes back online. Sequences are important because they eliminate replication conflicts by preventing duplicate primary key values across disconnected applications. All sequences must have a unique name.

You can accomplish this by modifying your sequence names by preceding them with your application name.

Figure 2–16 displays the Sequences tab.

Figure 2–16 Sequences Tab



As described in Table 2–10, the Sequences dialog includes the following fields.

Table 2–10 Sequences Dialog Description

Field	Description	Required
Name	The name of the sequence used by the Web-to-Go application in disconnected mode.	Yes
Type	The type of sequence used by the Web-to-Go application in disconnected mode. Window. The window sequence assigns a unique range of values to each client. Window sequences are unique to each client and never overlap with those of other clients. When a client uses all the values in its sequence range, Web-to-Go recreates the sequence with a new, unique range of values the next time the client goes offline.	Yes
Start Value	The sequence's start value on the Mobile Client for Web-to-Go. The sequence begins at this number and then increments according to the increment number you define.	Yes
Increment	The number by which the sequence increments on the Mobile Client for Web-to-Go, beginning at its start value.	Yes
Window Size	Defines the range of numbers in a window sequence.	Yes

Table 2–10 (Cont.) Sequences Dialog Description

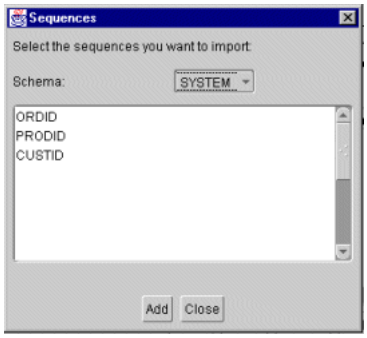
Field	Description	Required
Threshold	Defines the minimum range of required numbers in a window sequence. Web-to-Go creates a new sequence when the existing one reaches this range and when the client goes offline.	Yes
Server Start	The sequence's start value on the Oracle database. The sequence begins at this number and then increments according to the increment number you define. This number must be different from the sequence's start value on the Mobile Client for Web-to-Go.	No
Server Increment	The number by which the sequence increments on the Oracle database, beginning at its start value.	No
Server Minimum	The minimum start value for an ascending sequence on the Oracle database. For example, an ascending sequence could start at 1 and continue on in ascending order.	No
Server Maximum	The maximum start value for a descending sequence on the Oracle database. For example, a descending sequence could start at -1 and continue in descending order.	No

You can add or remove sequences from the Sequences dialog by clicking the Add or Remove button.

2.1.8.1 Importing Sequences

To import sequences from an Oracle database, click the Import button. As Figure 2–17 displays, the Sequences dialog appears.

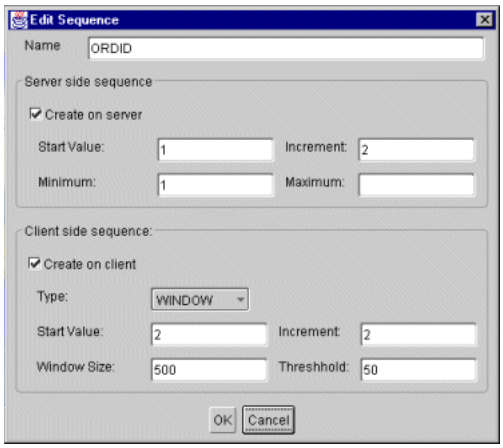
Figure 2–17 Sequences Dialog



Select the sequence you want to import, click Add, and then click Close.

To edit a sequence, select the sequence from the Sequences dialog and click Edit. As Figure 2–18 displays, the Edit Sequences dialog appears.

Figure 2–18 Edit Sequences Dialog



As Table 2–11 describes, edit the sequence by modifying the following features of the Edit Sequences dialog.

Table 2–11 Edit Sequences Dialog Description

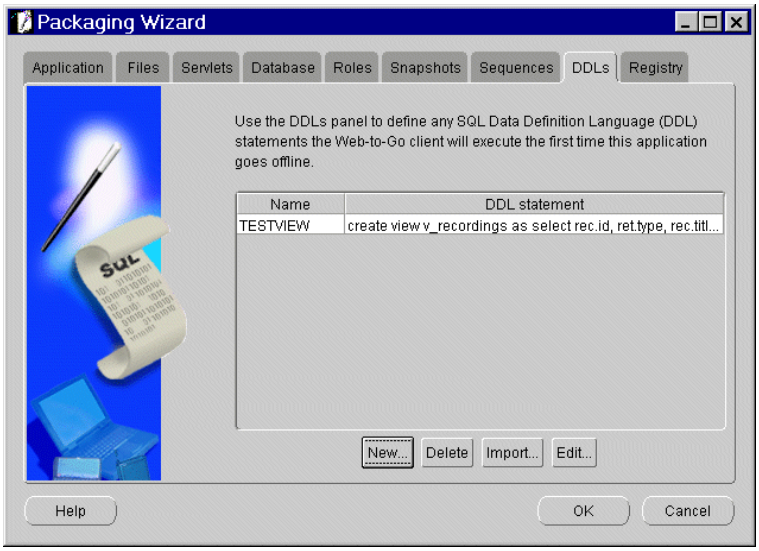
Feature	Description
Name	The name of the sequence.
Create on Server	When selected, this check box enables the options for creating a sequence on the Oracle database. Information entered by the user is used to generate a SQL script to create the sequence on the Oracle server.
Start Value	The start value of the sequence on the Oracle database.
Increment	The increment of the sequence on the Oracle database, beginning with its start value.
Minimum	The minimum start value for an ascending sequence on the Oracle database. For example, an ascending sequence could start at 1 and continue in ascending order.
Maximum	The maximum start value for a descending sequence on the Oracle database. For example, a descending sequence could start at -1 and continue in descending order.
Create on Client	When selected, this check box enables the options for creating a sequence on the Mobile Client for Web-to-Go.
Type	Defines the type of sequence on the Mobile Client for Web-to-Go. Options include the window and leapfrog sequences.
Start Value	The sequence start value on the Mobile Client for Web-to-Go.
Increment	The increment of the sequence on the Mobile Client for Web-to-Go, beginning with its start value.
Window Size	The range of numbers that constitute a window sequence on the Mobile Client for Web-to-Go. This information is not used by the leapfrog sequence.
Threshold	The minimum range of required numbers in a window sequence. Web-to-Go creates a new sequence when the existing one reaches this range and when the client goes offline. This information is not used by the leapfrog sequence.

2.1.9 Defining Application DDLs

Use the DDLs dialog to define any DDL (Data Definition Language) statements that the Web-to-Go application can execute the first time it goes offline. DDLs are only supported on Windows 32 and Windows CE platforms. All DDL statements must have a unique name and the weight must be specified for every DDL. One way to accomplish this is to modify your DDL names by preceding them with your application name. After you publish the application to the Mobile Server, you can create additional DDL statements using the Mobile Manager.

Figure 2–19 displays the DDLs dialog.

Figure 2–19 DDLs Dialog



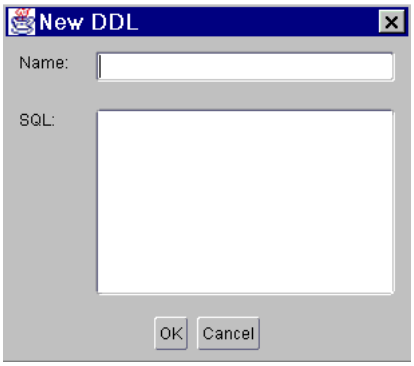
As described in Table 2–12, the DDLs dialog includes the following fields.

Table 2–12 DDLs Dialog Description

Field	Description
Name	The DDL name.
DDL Statement	Defines DDL statements with the Web-to-Go application. These DDL statements will be executed when the Web-to-Go application runs on the client.
Weight	The order of DDLs to be executed on the Mobile Client.

You can add or remove DDLs from the DDLs dialog by clicking the Add or Remove button. When you click the ADD button, the New DDL dialog appears, as described in Figure 2–20.

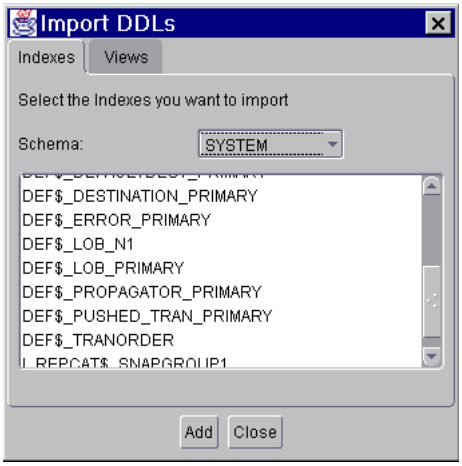
Figure 2–20 New DDL Dialog



2.1.9.1 Importing Views and Index Definitions

To import views and index definitions from an Oracle database, click the Import button. As displayed in Table 2–21, the Import DDLs dialog appears.

Figure 2–21 Import DDLs Dialog



To import an index definition, click the Indexes tab and then click the schema from which you want to import an index. Select the index you want to import, click Add, and then click Close.

To import a view definition, click the Views tab and then click the schema from which you want to import a view. Select the view you want to import, click Add, and then click Close.

2.1.10 Editing Applications

You can edit applications by launching the Packaging Wizard and selecting "Edit an existing application." Although you can manually create or edit applications by writing or modifying an XML document that conforms to the Mobile Server's DTD file, for best results, you should use the Packaging Wizard to create or modify applications.

2.1.11 Troubleshooting

The Packaging Wizard also supports development mode. In this mode, the Packaging Wizard only enables you to define web application information, list the application files, compile JSPs, add servlets, and make registry changes. Since the application is packaged to your local machine, it requires neither connectivity nor database information.

To launch the Packaging Wizard in development mode, enter the following using the Command Prompt.

```
wtgpack -d
```

2.2 Packaging Wizard Synchronization Support

The Packaging Wizard and the Mobile Manager provide the ability to perform the most commonly used functions of the publish and subscribe model, package and publish applications, create or drop users, and create or drop subscriptions. More sophisticated functionality is provided by the Consolidator and Resource Manager APIs. Table 2–13 describes basic features.

Table 2–13 Packaging Wizard Synchronization Support

Function	Packaging Wizard	Mobile Manager	API
Open Connection	No	No	Yes
Create User	No	Yes	Yes
Drop User	No	Yes	Yes
Create Publication	Yes	No	Yes
Create Publication Item	Yes	No	Yes
Create Publication Item Index	Yes	No	Yes
Drop Publication	No	Yes	Yes
Drop Publication Item	Special - See the Packaging Wizard documentation for more details.	No	Yes
Drop Publication Item Index	Yes	No	Yes
Create Sequence	Yes	No	Yes
Create Sequence Partition	Yes	No	Yes
Drop Sequence	Yes	No	Yes
Drop Sequence Partition	Yes	No	Yes
Add Publication Item	Yes	No	Yes
Remove Publication Item	No	No	Yes
Create Subscription	No	Yes	Yes
Deinstantiate Subscription	No	No	Yes
Set Subscription Parameter	No	Yes	Yes
Drop Subscription	No	Yes	Yes
Commit Transaction	No	No	Yes
Rollback Transaction	No	No	Yes

Table 2–13 (Cont.) Packaging Wizard Synchronization Support

Function	Packaging Wizard	Mobile Manager	API
Close Connection	No	No	Yes

More advanced features of Consolidator are only generally available by using the Consolidator and Resource Manager APIs. Table 2–14 describes these features.

Table 2–14 Consolidator Advanced Function Description

Function	Packaging Wizard	Mobile Manager	API
Create Virtual Primary Key Column	Yes	No	Yes
Drop Virtual Primary Key Column	Yes	No	Yes
Add Mobile DML Procedure	Yes	No	Yes
Remove Mobile DML Procedure	Yes	No	Yes
Reinstantiate Publication Item	No	No	Yes
Parent Hint	Yes	No	Yes
Dependency Hint	Yes	No	Yes
Remove Dependency Hint	Yes	No	Yes
Enable Publication Item Query Cache	No	No	Yes
Disable Publication Item Query Cache	No	No	Yes
Primary Key Hint	Yes	No	Yes
Purge Transaction	No	No	Yes
Execute Transaction	No	No	Yes
Complete Refresh	Yes	Yes	Yes
Execute Statement	No	No	Yes
Generate Metadata	No	No	Yes
Reset Cache	No	No	Yes
Cache Dependencies	No	No	Yes
Remove Cache Dependencies	No	No	Yes
Get Current Time	No	No	Yes
Authenticate	No	Yes	Yes
Set Restricting Predicate	No	No	Yes
Alter Publication	Yes	No	Yes

3

Database Tools and Utilities For Win32 and WinCE Platforms

This chapter describes how to use the following database utilities for the Windows 32 and Windows CE platforms. As described in Table 3–1, the utility names are listed in alphabetical order.

Table 3–1 Database Tools and Utilities

Utility	Description
`MSQL`	Allows users to execute SQL statements against the local database.
`ValidateDB`	Allows you to inspect and diagnose database corruptions.
`Support for Linguistic Sort`	Allows databases to be created with linguistic sort capability enabled.
`CREATEDB`	Use this to create Oracle Database Lite databases.
`DECRYPDB`	Use this to decrypt your Oracle Database Lite databases.
`dropjava`	This is a command-line utility you can use to remove Java classes from Oracle Database Lite. For more information, see the *Oracle Database Lite Developer's Guide for Java*.
`ENCRYPDB`	Use this to encrypt your Oracle Database Lite.
`loadjava`	This is a command-line utility you can use to load a Java class into Oracle Database Lite. For more information, see the *Oracle Database Lite Developer's Guide for Java*.
`MIGRATE`	Use this to migrate to Oracle Database Lite from a previous release.
`Mobile SQL`	Mobile SQL is a command line interface that allows you to connect to Oracle Database Lite databases.
`ODBC Administrator and the Oracle Lite ODBC Driver`	Use this to manage ODBC connections by creating data source names (DSNs) that associate the Oracle Database Lite ODBC Driver with the Oracle Database Lite that you want to access through the driver.
`ODBINFO`	Use this utility to find out the version number and volume ID of an Oracle Database Lite database.
`OLLOAD`	Use this command-line tool to load data from an external file into a table in Oracle Database Lite, or to unload (dump) data from a table in Oracle Database Lite to an external file.
`REMOVEDB`	Use this to remove Oracle Database Lite databases.
`VALIDATEDB`	Use this to validate the structure of an Oracle Lite database.

3.1 MQSL

Mobile SQL is a GUI-based application that runs on the client device (laptop, Palm OS, and Windows CE). It allows the user to execute SQL statements against the local database. It is both a developers tool and a code example. It allows users to access functionality provided by the ODBC and Oracle Database Lite OKAPI interfaces of the underlying Oracle Database Lite database engine.

Mobile SQL allows you to create, access, and manipulate Oracle Database Lite on Palm Computing platform devices. Using Mobile SQL you can accomplish the following:

- Create databases
- View tables
- Execute SQL statements

The following sections describe how to use the MSQL tool. Topics include:

- Section 3.1.1, "MSQL for Windows 32"
- Section 3.1.2, "MSQL for Windows CE"

3.1.1 MSQL for Windows 32

The following sections describe how to populate your database using Mobile SQL and use Mobile SQL. Topics include:

- Section 3.1.1.1, "Populating your Database Using Mobile SQL"
- Section 3.1.1.2, "Mobile SQL"

3.1.1.1 Populating your Database Using Mobile SQL

You can use SQL scripts to create tables and schema, and to insert data into tables. A SQL script is a text file, generally with a .sql extension, that contains SQL commands. You can run a SQL script from the Mobile SQL prompt by typing:

```
SQL> @<ORACLE_HOME>\DBS\Poldemo.sql
```

You can also type:

```
SQL> START <filename>
```

Note: You do not need to include the .sqlz file extension when running the script.

3.1.1.2 Mobile SQL

Mobile SQL is an application that runs as a command line interface. It allows the user to execute SQL statements against the local database. It is both a developers tool and a code example. It allows users to access functionality provided by the ODBC and Oracle Database Lite OKAPI interfaces of the underlying Oracle Database Lite database engine.

The following sections describe information relevant to database access, starting Mobile SQL and Mobile SQL commands. Topics include:

- Section 3.1.1.3, "Database Access"
- Section 3.1.1.4, "Starting Mobile SQL"
- Section 3.1.1.5, "SET TERM {ON|OFF}"

- Section 3.1.1.6, "SET TIMING {ON | OFF}"
- Section 3.1.1.7, "SET VERIFY {ON | OFF}"

3.1.1.3 Database Access

Mobile SQL accesses the database through both the ODBC and OKAPI interface. Most functions are performed through ODBC, but functions that ODBC cannot handle are implemented using OKAPI function calls.

3.1.1.4 Starting Mobile SQL

Mobile SQL is started by opening the Oracle_Home\Mobile\SDK\Bin directory and double-clicking on the msql.exe file. This starts the command-line interface which accepts standard SQL commands. For more information, see the *Oracle Database Lite SQL Reference*.

3.1.1.5 SET TERM {ON|OFF}

Controls the display of output generated by commands executed from a script. OFF suppresses the display so that you can spool output from a script without seeing the output on the screen. ON displays the output. TERM OFF does not affect output from commands you enter interactively.

3.1.1.6 SET TIMING {ON|OFF}

Controls the display of timing statistics. ON displays timing statistics on each SQL command. OFF suppresses timing of each command.

3.1.1.7 SET VERIFY {ON|OFF}

Controls whether to list the text of a SQL statement or PL/SQL command before and after replacing substitution variables with values. ON lists the text; OFF suppresses the listing.

3.1.2 MSQL for Windows CE

The Oracle Database Lite database format is the same for Windows 32 and Windows CE. You can create and test your snapshots on Windows 32 using the Windows 32 MSQL command line. You can then copy the database to the Windows CE platform. Use the Windows CE MSQL to manipulate the database that is on your device.

The following sections enable you to use the Mobile SQL application. Topics include:

- Section 3.1.2.1, "Using Mobile SQL"

3.1.2.1 Using Mobile SQL

Mobile SQL is an application that runs as a command line interface. It allows the user to execute SQL statements against the local database. It allows users to access functionality provided by the interfaces of the underlying Oracle Database Lite database engine.

The following sections describe how to access the database and start Mobile SQL. Topics include:

- Section 3.1.2.2, "Database Access"
- Section 3.1.2.3, "Starting Mobile SQL"

3.1.2.2 Database Access

Mobile SQL accesses the database through both the ODBC and OKAPI interface. Most functions are performed through ODBC, but functions that ODBC cannot handle are implemented using OKAPI function calls.

3.1.2.3 Starting Mobile SQL

Mobile SQL is started by opening the `<Oracle_home>\Mobile\Sdk\WinCE`, select the folder representing the version Windows CE, then the processor on your device. Double-click on the mSQL.exe file. This starts the GUI which accepts standard SQL commands. For more information, see the *Oracle Database Lite SQL Reference*.

3.2 Database Tools and Utilities for PALM

This section describes how to use the ValidateDB utility. Topics include:

- Section 3.2.1, "Overview"
- Section 3.2.2, "Installing ValidateDB"
- Section 3.2.3, "Running ValidateDB"
- Section 3.2.4, "Sending Corrupted Databases"
- Section 3.2.5, "Must Send Databases"

3.2.1 Overview

In some cases, the Oracle Database Lite database on Palm may become corrupted. It can be caused by hardware problems or bugs in the database code. Using the ValidateDB utility, you can inspect and diagnose database corruptions. As database users and application developers, you can run the ValidateDB utility to check the database for consistency. The Oracle Database Lite development group then uses the validateDB utility to diagnose the extent of corruption and fixes the problem.

3.2.2 Installing ValidateDB

To install the validateDB utility, you just need to install a single file named `validatedb.prc`. It is included on your system during installation and is located in the directory named `Lite\Runtime`. Using the HotSync application, you must install this file on your Palm device or emulator (using the right-click menu).

3.2.3 Running ValidateDB

To run the validateDB utility, perform the following steps.

1. Click the ValidateDB icon. The main ValidateDB form appears. As Table 3–2 describes, the ValidateDB form contains the following items.

Table 3–2 ValidateDB Form Description

Item	Description
Oracle Database Lite database list	List of Oracle Database Lite databases installed on the device (or emulator).
Log to Desktop	If selected, logs database information to your desktop.

2. To validate a database, choose the required Oracle Database Lite database from the list displayed. To validate all databases, click **Validate All**.

> **Note:** You will be prompted for the password of each encrypted database that requires validation.

After validating the chosen database(s), the ValidateDB utility displays the following alerts:

- No Errors Found - This alert indicates that no corruption has been detected.
- CorruptedDB - This alert indicates that the utility has detected some corruption and the databases need to be sent to the Oracle Database Lite development group for further investigation.
- System Fatal - This alert indicates that the utility has detected system fatal alerts and the databases need to be sent to the Oracle Database Lite development group for further investigation. You should reset the device if you receive a system fatal alert.
- Assertion - This alert indicates that the utility has detected some errors and the databases need to be sent to Oracle Database Lite development group for further investigation.

> **Note:** Do not select the "Log to Desktop" box as it is used primarily by Oracle Database Lite developers to log further debugging information to a desktop computer. If you select this box, the validateDB utility stops functioning and does not respond.

3.2.4 Sending Corrupted Databases

If the validateDB utility detects corruption in the database, it is mandatory that you send all such databases to the Oracle Database Lite development group. To send corrupt databases, click **BackupAll**. This command sets up the backup flag for all Oracle Database Lite databases on the device. During the next HotSync instance, the Oracle Database Lite databases are backed up on the desktop computer. After running the HotSync application, you will find these databases in the directory of the HotSync manager named `Palm\`(HotSync user name)`\Backup`.

3.2.5 Must Send Databases

This section lists databases that must be sent along with the database that has been detected as corrupt.

1. All databases which appear in the list on the validateDB form.
2. Databases which start with the same name as the ones on the list, but contain the extension $1, $2, ...,. For example OrdersODB$1.PDB, OrdersODB$2.PDB, ... These are Oracle Database Lite extensions for large databases.
3. okSysDB.PDB
4. okTransLog.PDB

To fix corrupted databases, the Oracle Database Lite development group retrieves further debugging information using the validateDB utility during the problem diagnosis phase.

3.3 CREATEDB

Description

Utility for creating a database.

Syntax

CREATEDB DataSourceName DatabaseName [[[VolID] DATABASE_SIZE] EXTENT_SIZE] [collation sequence]

Keywords and Parameters

DataSourceName

Data source name, used to look up the `ODBC.TXT` file for the default database directory.

> **Note:** If you specify an invalid DSN, Oracle Database Lite ignores the DSN and creates the database in the current directory. To access this database through ODBC, you must create a DSN for the database that points to the directory in which the database resides. For instructions on adding a DSN, see Section 3.7.1, "Adding a DSN Using the ODBC Administrator".

DatabaseName

Name of the database to be created. It can be a full path name or just the database name. If only the database name is given, the database is created under the Data Directory for the data source name specified in the `ODBC.TXT` file. The extension for the database name must always be `.ODB`. If a name without the `.ODB` is given, the `.ODB` is appended.

VolID

When specified, the `VolID` is used as the database ID, instead of the database ID from the `POLITE.INI` file. The ID must be unique for each database.

DATABASE_SIZE

The database size in bytes.

EXTENT_SIZE

An incremental amount of pages in a database file. When a database runs out of pages in the current file, it extends the file by this number of pages.

COLLATION_SEQUENCE

This parameter is a string constant which creates the database as enabled for linguistic sorting when a value other than the default is used. A collation sequence specified here overrides a collation sequence set using the `NLS_SORT [collation_sequence]` parameter in the `polite.ini` file. The string can also be one of the options listed in Table 3–3:

Table 3–3 Collation Sequence Values

Collation Sequence	Description
`BINARY`	Default. Two strings are compared character by character and the characters are compared using their binary code value.

Table 3–3 (Cont.) Collation Sequence Values

Collation Sequence	Description
FRENCH	Two strings are compared according to the collation sequence of French. Supported by ISO 8859-1 or IBM-1252.
GERMAN	Two strings are compared according to the collation sequence of German. Supported by ISO 8859-1 or IBM-1252.
CZECH	Two strings are compared according to the collation sequence of Czech. Supported by ISO 8859-2 or IBM-1250.
XCZECH	Two strings are compared according to the collation sequence of Xczech. Supported by ISO 8859-2 or IBM-1250.

Note: There is no way to alter a collation sequence after the database is created.

Examples

```
createdb polite db1

createdb polite c:\testdir\db2.odb 300

createdb polite polite french
```

3.4 DECRYPDB

Description

This tool allows you to decrypt an encrypted Oracle Database Lite. For more information, see Section 3.5, "ENCRYPDB".

SYNTAX

DECRYPDB *DSN* | NONE *DBName* [*Password*]

Keywords and Parameters

DSN

Data Source Name of Oracle Database Lite that you want to decrypt. If you specify NONE, you must the enter the DBName with the full path name (without the .ODB extension).

DBName

Name of the database to be decrypted. If DSN was specified as NONE, the DBName must be entered with the full path name.

Password

Optional. The password used previously to encrypt Oracle Database Lite. If you do not enter the password, DECRYPDB prompts you to enter it.

Comments

An Oracle Database Lite database cannot be decrypted if there is any open connection to the database.

If you call this utility from another program, the possible values returned are listed in Table 3–4:

Table 3–4 DECRYPDB Return Codes

Return Code	Description
EXIT_SUCCESS	Success
EXIT_USAGE	Command line arguments are not properly used or are in error
EXIT_PATH_TOO_LONG	Path is too long
EXIT_SYSCALL	I/O error while making new decrypted copy on disk
EXIT_BAD_PASSWD	Incorrect password supplied

For more information, see the comments in Section 3.5, "ENCRYPDB".

3.5 ENCRYPDB

Description

This tool allows you to encrypt Oracle Database Lite with a password and to change a database password. The password prevents unauthorized access to the database and encrypts the database, so that the data stored in the database files cannot be interpreted. For more information, see Section 3.4, "DECRYPDB".

ENCRYPDB uses CAST5 encryption, which is a 128-bit, DES compliant encryption scheme.

Syntax

ENCRYPDB *DSN* | NONE *DBName* [*New_Password* [*Old_Password*]]

Keywords and Parameters

DSN

Data Source Name of Oracle Database Lite that you want to encrypt. If you specify NONE, DBName must be a fully qualified database name with the full path name (without the .ODB extension). If the DSN is a value other than NONE, then the name must appear as a data source name in the ODBC.TXT file.

DBName

Name of the database to be encrypted. If DSN was specified as NONE, DBName must be entered with the full path name.

New_Password and Old_Password

Optional, the password (or previously used password) for encrypting the database. This password can be 128 characters in length. If you do not enter a password, ENCRYPDB prompts you to enter one. Since both passwords are optional in the command line to invoke the utility, the command line could have three different forms:

- No password given: If the database is already encrypted, then ENCRYPDB assumes that the user is trying to change the password of the database. It prompts the user for the old password once and new password twice, and encrypts the database using the new password. If the database is not already encrypted, ENCRYPDB prompts for the new password twice and encrypts the database using this new password.

- One password given: This password is assumed to be the new password. If the database is already encrypted, ENCRYPDB prompts for the old password and encrypts the database using the new password.
- Both passwords given: ENCRYPDB assumes that the first password is the new password and the second is the old password.

Comments

If you call this utility from another program, the possible values returned are listed in Table 3–5:

Table 3–5 ENCRYPDB Return Codes

Return Code	Description
EXIT_SUCCESS	Success
EXIT_USAGE	Command line arguments are not properly used or are in error
EXIT_PATH_TOO_LONG	Path is too long
EXIT_SYSCALL	I/O error while making new encrypted copy on disk
EXIT_BAD_PASSWD	Incorrect password supplied

The default Oracle Database Lite (POLITE.ODB) is not encrypted. After encrypting an Oracle Database Lite, every user that attempts to establish a connection to the encrypted Oracle Database Lite must provide the valid password. If the password is not provided, Oracle Database Lite returns an error. An Oracle Database Lite database cannot be encrypted if there are any open connections to the database.

You should consider the following when encrypting and decrypting Oracle Database Lite:

- You cannot decrypt an encrypted database without the password. Make sure you back up your database in a secure place before you encrypt it. Another user of the same database can create a copy with a new user name for a user who loses their password, otherwise, there is no method to recover a database where the passwords are lost.
- After encrypting the database, you must include the password in the connect string to connect to the database.
- A password encrypts the entire database. It is not a user-specific password.
- Database encryption does not prevent a third party from removing an Oracle Lite Database. That is, removedb and rmdb remove a database without checking the password. Use tools that protect unauthorized users from manipulating your file system.
- ODBC applications that connect to an encrypted Oracle Database Lite database need to specify a valid password. It is customary to prompt for the password at runtime rather than to code it in the application. Most ODBC applications can use the SQLDriverConnect function with the DRIVER= option, rather than the SQLConnect function, if the applications require the Oracle Database Lite ODBC driver to prompt for the password at runtime.
- All sample applications provided with this release of Oracle Database Lite are designed to run against a database that is not encrypted.
- You can use DECRYPDB and ENCRYPDB (in this order) to change the password of a database. However, DECRYPDB creates an Oracle Database Lite database in plain

text before ENCRYPDB encrypts it. This results in a database in plain text form, for a short period of time, and is not recommended.

- For encrypted databases, all user names and passwords are written to a file named DSN.OPW. Each user can then use the password as a "key" to unlock the .OPW file before the .ODB file is accessed. When you copy or back up the database, you should include the .OPW file.

3.5.1 Synchronizing with an Encrypted Database

These steps are required to synchronize with an encrypted Oracle Database Lite database.

1. Retrieve the user password from the Mobile Server Repository.
2. Convert the password into uppercase. For example, change "manager" into "MANAGER".
3. Launch Mobile Sync (msync.exe) and execute a sync. Provide username, password and Mobile Server URL. Select Apply and then select Sync. This creates an non-encrypted Oracle Database Lite database.
4. Encrypt the Oracle Database Lite database using the ENCRYPDB utility. Enter the converted uppercase password, for example MANAGER.
5. Continue to synchronize.

3.6 MIGRATE

Description

Utility for migrating a database from a previous version of Oracle Lite to Oracle Database Lite 10*g*. The utility migrates your Oracle Database Lite 3.6 database and makes a backup copy with a .36 extension. If you have an earlier release of Oracle Database Lite, see the *Oracle Database Lite Installation and Configuration Guide for Windows NT/2000/XP* for more information.

Before you use this utility, you must install the current release of Oracle Database Lite. Also, if your database is encrypted, you must first decrypt it before using this utility.

Syntax

MIGRATE DSN DBName

where DB Name can be the database name or the database path and name.

Keywords and Parameters

DSN

Data source name of the database to migrate. This is used to look up the default database directory in the ODBC.INI file for the database name given in DBName. If the DSN has the value NONE the DBName should be a complete pathname of the database file.

DBName

The database name, or the path and database name, to migrate. If only the database name is specified, the database file must exist in the directory specified in the DataDirectory parameter (under the data source name) in the ODBC.INI file.

Comments

As mentioned in this section, you must install Oracle Database Lite before you use this utility.

Any messages generated by the `MIGRATE` utility are displayed on the screen in the command window.

Using this utility allows you to compress empty space in your existing Oracle Database Lite database.

This utility does not support the migration of java stored procedures.

Examples

```
MIGRATE polite db1

MIGRATE none c:\testdir\db1.odb
```

3.7 ODBC Administrator and the Oracle Database Lite ODBC Driver

A Data Source Name (DSN) associates the Oracle Database Lite ODBC Driver with the Oracle Database Lite database that you want to access through the driver. The Oracle Database Lite installation process creates a default `DSN`, `POLITE`, for the Oracle Database Lite database. You can also create additional DSNs for the additional Oracle Database Lite databases that you create.

Microsoft provides the ODBC Administrator, a tool for managing the `ODBC.INI` file and associated registry entries in `Windows 98/NT/2000/XP`. The `ODBC.INI` file and the Windows registry store the DSN entries captured through the ODBC Administrator. Using the ODBC Administrator, you can relate a DSN to the Oracle Database Lite ODBC Driver.

Note: This document does not provide instructions on using the ODBC Administrator. See its online help for this information.

In the ODBC Administrator, in addition to the DSN, you must specify the parameters listed in Table 3–6:

Table 3–6 ODBC Administrator DSN Parameters

DSN Parameter	Description
`Data Description`	An optional description for the data source.
`Database Directory`	The path to the data directory where the database resides. This is an existing path.
`Database`	Oracle Database Lite database name to be created. Do not include the `.ODB` extension.
`Default Isolation Level`	Determines the degree to which operations in different transactions are visible to each other. For more information on the supported isolation levels, refer the *Oracle Database Lite Developer's Guide*. The default level is "`Read Committed`".

Table 3–6 (Cont.) ODBC Administrator DSN Parameters

DSN Parameter	Description
`Autocommit`	Commits every database update operation in a transaction when that operation is performed. Autocommit values are Off and On. The default value is Off. **Note**: In the Microsoft ODBC SDK, the ODBC driver defaults to auto-commit mode. However, the default for Oracle Database Lite is manual-commit mode. In this environment, if you execute SQLEndTrans / SQLTransact call with SQL_COMMIT option using the ODBC driver, you receive a SQL_SUCCESS, because ODBC believes that auto-commit is on. However, no commit actually occurs, because ODBC transfers the transaction to Oracle Database Lite, whose default is manual-commit. You must configure the Microsoft ODBC Driver Manager to transfer control of the SQLEndTrans / SQLTransact API call to Oracle Database Lite by explicitly setting autocommit to OFF in ODBC. When you do this, ODBC does not try to autocommit, but gives control of the transaction to Oracle Database Lite. To set auto-commit to off, execute either the SQLSetConnectAtrr or SQLSetConnectOption method with SQL_AUTOCOMMIT_OFF as the value of the SQL_AUTOCOMMIT option. Then, the SQLEndTrans / SQLTransact calls will commit as defaulted within Oracle Database Lite. Thus, if you want auto-commit on, turn it on only within Oracle Database Lite.
`Default Cursor Type`	■ *Forward Only*: Default. A non-scrollable cursor which only moves forward but not backward through the result set. As a result, the cursor cannot go back to previously fetched rows. ■ *Dynamic*: Capable of detecting changes to the membership, order, or values of a result set after the cursor is opened. If a dynamic cursor fetches rows that are subsequently deleted or updated by another application, it detects those changes when it fetches those rows again. ■ *Keyset Driven*: Does not detect change to the membership or order of a result set, but detects changes to the values of rows in the result set. ■ *Static*: Does not detect changes to the membership, order or values of a result set after the cursor is opened. If a static cursor fetches a row that is subsequently updated by another application, it does not detect the changes even if it fetches the row again.

For example, the DSN entry for `POLITE` in the **`ODBC.INI`** file may contain:

```
[POLITE]

Description=Oracle Lite Data Source

DataDirectory=C:\ORANT\OLDB40

Database=POLITE

IsolationLevel=Repeatable Read

CursorType=Dynamic
```

3.7.1 Adding a DSN Using the ODBC Administrator

To add a DSN using the ODBC Administrator:

1. Start the ODBC Administrator, either by selecting its icon in the Oracle Database Lite program group, or by typing the following at a DOS prompt:

   ```
   C:\>ODBCAD32
   ```

2. Click Add.
3. Double-click the Oracle Database Lite *nn* ODBC Driver (*nn* is the release number) from the list of Installed ODBC Drivers.
4. Next, add the DSN name and define the parameters in the ODBC driver setup dialog. Refer the preceding table for help in defining the parameters.

3.7.2 Adding a DSN which points to Read-Only Media (CD-ROM)

1. Create the DSN as explained in Section 3.7.1, "Adding a DSN Using the ODBC Administrator".
2. Add the following line to the new DSN in the **`ODBC.INI`** file:

   ```
   ReadOnly = True
   ```

Note: You can define a DSN which points to a file on a CD-ROM. Simply point the DSN to the CD-ROM drive and directory and provide the file name of the database file. Then modify the **`ODBC.INI`** file to add the line "`ReadOnly=True`" to the data source definition. ODBC programmers can call the following before opening the database to enable this feature (instead of adding the line to the ODBC.INI file):

```
SQLSetConnectOption( hdbc, SQL_ACCESS_MODE, SQL_
MODE_READ_ONLY )
```

Setting a database file to read-only suppresses the creation of log files. Updates, insertions, deletions, or commits appear to work on the in-memory image of tables. However, when you commit, these changes are not written to the database file. If you exit your application, reconnect, and issue your query, you see your original data.

3.8 ODBINFO

Description

You can use `ODBINFO` to find out the version number and volume ID of an Oracle Database Lite database. `ODBINFO` can also display and set several parameters.

Syntax

To display current information without making any changes use the syntax:

```
odbinfo [-p passwd]DSN DBName
```

You can also use:

```
odbinfo [-p passwd] NONE dbpath\dbanme.odb
```

For example:

```
odbinfo -p tiger polite polite
```

```
odbinfo NONE c:\orant\oldb40\polite.odb
```

If your database is encrypted you need to include the password.

Parameters

To set or clear parameters, use one or more "+" or "-" parameter arguments before the DSN or NONE. For example:

```
odbinfo +reuseoid -pagelog -fsync polite polite
```

You can use the parameters listed in Table 3–7 with the `ODBINFO` utility:

Table 3–7 ODBINFO Parameters

Parameter	Description
`pagelog`	By default, a commit backs up modified database pages to *filename.plg* before actually writing the changes to *filename.odb*. If an application or the operating system experiences a failure during a commit, the transaction is cleanly rolled back during the next connect. If `-pagelog` is specified, no backup is created and the database can become corrupted if a failure occurs.
`fsync`	Oracle Database Lite generally forces the operating system to write all the modified buffers associated with the database back to disk during a commit. If this option is disabled (`-fsync`), the operating system can keep the changes in memory until a later time. If the system (but not the application) crashes before the buffers are flushed, the database can become corrupted. Using `odbinfo -fsync -pagelog` improves the performance of applications that use many small transactions (with autocommit on) or ones with massive updates. However, if the database is corrupted, there is no straightforward way to repair it or recover the data. Therefore these two options should only be cleared during initial loading of the database, if (1) the **.ODB** file is backed up on regular basis, or (2) the data in the database can be recovered from some other source. Using this option has no effect on applications that seldom update the database. Setting the transaction isolation level to `SINGLE USER` has more impact in this case.
`reuseoid`	By default, Oracle Database Lite does not reuse the `ROWID` of any row that exists in a table until the table is dropped. The "Slot Deleted" error is returned when accessing a deleted object. This uses two bytes of storage for each deleted object, causing performance and disk space usage to degrade over time if rows are constantly inserted and deleted. If you use `odbinfo +reuseoid`, new rows can reuse `ROWIDs` of previously deleted rows. However, this may not free all the space in a table that already has many deleted objects. For best results, you should set this option immediately after you create your database. This option is safe for pure relational applications. However, SQL applications that use `ROWID` and `OKAPI` applications that use direct pointers between objects need to verify that all references to an object are set to `NULL` before the object is deleted. Otherwise, dangling references may eventually point to some other, unrelated object.

Table 3–7 (Cont.) ODBINFO Parameters

Parameter	Description
`compress`	This option (which is "on" by default) enables run-length compression of objects. Run-length compression takes very little CPU time, so you should only deselect (`-compress`) this option if: ■ Operating system-level file compression is used, such as `DriveSpace` or a `NTFS` compressed attribute. In this case not compressing the same data twice provides a better compression ratio. ■ Most objects in the database are frequently updated to a highly compressible state (for example, all columns set to `NULL`), and the data cannot be compressed well (such as binary columns with random data). In these cases, using this option (`+compress`) can result in highly fragmented tables. Changing this option does not compress or decompress any existing objects in the database.

3.9 OLLOAD

For information on the Oracle Database Lite Load APIs, refer the *Oracle Database Lite Developer's Guide*.

Description

This command line tool allows you to load data from an external file into a table in Oracle Database Lite, or to unload (`dump`) data from a table in the Oracle Database Lite database to an external file. Unlike SQL*Loader, `OLLOAD` does not use a control file you supply all data parameters and format information on the command line.

When loading data, `OLLOAD` takes an input file that contains one record per line with a separator character between fields. The default field separator is a comma (,). The records can also include fields with values that are quoted strings. The default is a single quote ('). See Data Parsing for more information on data parsing.

Syntax

To load a datafile:

```
olload [options] -load dbpath tbl [col1 col2 ...] [<datafile]
```

To unload (dump) to an outfile

```
olload [options] -dump dbpath tbl [col1 col2 ...] [<outfile]
```

Keywords and Parameters

[options]

See Options for a list of options.

-load

To use the load utility.

-dump

To use the unload (dump) utility.

dbpath

The path to the Oracle Database Lite file (.ODB file).

tbl

The table name. OLLOAD first attempts to find a table name in the user-specified case. If this fails, it searches for the uppercase of the user-specified name.

> **Note:** The default user is "SYSTEM". If you want to specify an OLLOAD operation for another user name's tables, prefix the tbl parameter with the user name and a dot (.).

col1 col2

The column name(s). OLLOAD first attempts to find a column name in the user-specified case. If this fails, it searches for the uppercase of the user-specified name.

[datafile] [outfile]

The source or destination file for the load or unload (dump) operations. If you do not specify a datafile or outfile, OLLOAD displays the output on the screen.

Options

-sep *character*

The field separator. If you do not specify this option, OLLOAD assumes that the separator character is a comma (,).

-quote *character*

The quote character. If you do not specify this option, OLLOAD assumes that the quote character is a single quote (').

-file *filename*

Use this option when loading and unloading data to specify the source or destination file name. When loading data, *filename* specifies the source file to load into the Oracle Database Lite database. When unloading (dumping) data, it is the destination file for the unloaded data.

> **Important:** To unload data from an Oracle Database Lite and load (or pipe) it to another Oracle Database Lite database, do not specify a file name for this option. See the second example in Examples for sample syntax.

-log *logfile*

Specify this option if you want to produce a log file listing rows that OLLOAD could not insert during load. If you do not specify a log file, loading stops at the first error.

-passwd *passwd*

The connection password for an encrypted database. You need to supply this password so that loading and unloading can occur.

-nosingle

Specify this option when you do not want to use single user mode. This degrades performance but allows other connections to the database.

-readonly

Specify this option when *unloading* data from a read-only Oracle Database Lite database, for example, one located on a CD-ROM.

-commit *count*

Use this option if you want `OLLLOAD` to commit after processing a specified number of rows. The default is `10000`. `OLLLOAD` prints an asterisk (`*`) to the screen each time it commits the specified number of rows. To disable the commit operation specify 0.

-mark *count*

Use this option if you want `OLLOAD` to print a dot on the screen after processing the specified number of records. The default is `1000`. To disable this feature specify 0.

Comments

Data Parsing

Table 3–8 shows examples for `OLLOAD` data parsing:

Table 3–8 Data Parsing Examples

Input	Data	Explanation
'Redwood Shores, CA'	Redwood Shores, CA	Enclosing the input string in quotes preserves spaces and punctuation within the string.
'O"Brien'	O'Brien	Represent a single quote with its escape sequence, two single quotes.
fire fly	firefly	Spaces in data that is not quoted are ignored.
,	NULL,NULL	Empty fields are NULL.
1,,3,	1,NULL,3,NULL	Empty fields are NULL.
	[no row inserted]	Completely empty lines are ignored.

If there are more values than database columns, extra values are ignored. Any missing values at the end of the line are set to `NULL`.

OLLOAD Utility Restrictions

OLLOAD does not support tab-delimited input files and `LONG` datatypes.

Examples

```
olload -quote \" -file p_kakaku.csv -load c:\orant\oldb40\polite.odb skkm01

olload -dump c:\orant\oldb40\polite.odb emp empno ename | olload -load myfile.odb
myemp
```

3.10 REMOVEDB

Description

Utility for deleting a database.

Syntax

REMOVEDB DataSourceName Database Name

Keywords and Parameters

DataSourceName

Data source name of the database you want to remove. The DSN can be a dummy argument such as none, in which case the database name must be a fully qualified filename.

DatabaseName

The name of the database to delete. It can be a full path name or just the database name. If only the database name is given, the database is deleted from the Data Directory for the data source name specified in the **`ODBC.INI`** file.

Examples

```
removedb polite db1
```

```
removedb none c:\testdir\db2.odb
```

3.11 VALIDATEDB

Description

This command-line tool validates the structures within the database file and if the database structure is found to be corrupted, lists the errors found in a file designated by the user. The tool checks the following:

- Objects - Header information for database objects. Flags are checked for consistency in case the object was moved or compressed. Object length is checked against a valid range. If the object is a BLOB, the object's frames are checked against the volume page bitmap.
- Index page entries - Checks that the creation of an index page entry results in the correct number of nodes or list of object identifiers.
- Index pages - Checks that all key values on the page are sorted. All objects contained on the page are validated. Page descriptor information such as the number of objects, the number of free bytes, and the number of entries are checked against the actual objects on the page.
- Groups - As each page is validated, the group descriptor information is checked against the actual number of pages and objects.
- Indexes - All the pages are validated against the btree. The tool also validates all page pointers. All levels of the btree are checked to validate that key values are in the sorted order as a whole. For leaf elements of the btree, all OIDs from the leaf page entries are checked for consistency with the actual group objects.

Syntax

validatedb *DSName DBName* [-p *password*] [-t *schemaname.tablename*] -file *outputfilename*

Keywords and Parameters

DSName

The data source name. This can also be NONE if no DSN is present.

DBName

If there is a DSN present, this is the database file name (without the .odb extension) if it is different from the default filename for the DSN. If there is no DSN, then VALIDATEDB uses the current directory unless the full path is specified. If there is a log file in the same directory as the database file, it is also validated.

password

Password for an encrypted database.

schemaname

Optional schema name. The default schema name is used unless this is specified.

tablename

Optional table name. The specified table is validated along with all of its indexes. If no table name is specified, the entire database is validated.

outputfilename

Optional filename for the text file where all errors and other related information revealed by VALIDATEDB are saved. The default is stdout.

Examples

```
validatedb polite polite -t emp -file out.txt
```

4

Load Utility

This document describes the Oracle Database Lite Load utility. Topics include:

- Section 4.1, "Overview"
- Section 4.2, "Syntax"
- Section 4.3, "Keywords and Parameters"

4.1 Overview

The Oracle Database Lite Load Utility is a command line tool which enables you to load data a from an external file into a table in Oracle Database Lite or to unload (dump) data from a table in Oracle Database Lite to an external file. Unlike SQL*Loader, OLLOAD does not use a control file in which you supply all data parameters and format information on the Command line.

When loading data, OLLOAD takes an input file that contains one record per line with a separator character between fields. The default field separator is a comma (,). These records can also include fields with values that are quoted strings. The default value is single quote (’). For more information on data parsing, see Section 4.3.1.1, "Comments".

4.2 Syntax

Loading a Datafile

To load a datafile, use the following syntax.

olload [options] -load *dbpath tbl* [*col1 col2 ...*] [<*datafile*]

Unloading (dump) to an Outfile

olload [options] -dump *dbpath tbl* [col1 col2 ...] [>*outfile*]

4.3 Keywords and Parameters

This section describes keywords and parameters that are available for the OLLOAD utility.

[options]

For a list of Options, see Section 4.3.1, "Options".

-load

To use the load utility.

-dump

To use the unload (dump) utility.

dbpath

The path to the Oracle Database Lite (`.odb`) file.

tbl

The table name. OLLOAD first attempts to find a table name in the user-specified case. If this fails, it searches for the upper-case of the user-specified name.

> **Note:** The default user is "SYSTEM". To specify an OLLOAD operation for another user name's tables, prefix the `tbl` parameter with the user name and a dot (.).

col1 col2

The column names. OLLOAD first attempts to find a column name in the user-specified case. If this fails, it searches for the upper-case of the user-specified name.

[datafile] [outfile]

The source or destination file for the load or unload operations. If you do not specify a datafile or outfile, OLLOAD displays the output on the screen.

4.3.1 Options

This section describes keyword and parameter options that are available for the OLLOAD utility.

-sep *character*

The field separator. If you do not specify this option, OLLOAD assumes that the separator character is a comma (,).

-quote *character*

The quote character. If you do not specify this option, OLLOAD assumes that the quote character is a single quote (').

-file *filename*

Use this option when loading and unloading data to specify the source or destination file name. When loading data, filename specifies the source file to load into Oracle Database Lite. When unloading (dumping) data, it is the destination file for the unloaded data.

> **Note:** To unload data from Oracle Database Lite and load (or pipe) it to another Oracle Database Lite, do not specify a file name for this option. For a description of sample syntax, see Section , "Examples".

-log *logfile*

Specify this option if you want to produce a log file listing rows that OLLOAD could not insert during load. If you do not specify a log file, loading stops at the first error.

-passwd *passwd*

The connection password for an encrypted database. You need to supply this password so that loading and unloading can occur.

-nosingle

Specify this option when you do not want to use single user mode. This degrades performance but allows other connections to the database.

-readonly

Specify this option when unloading data from a read-only Oracle Database Lite, for example, one located on a CD-ROM.

-commit *count*

Use this option if you want OLLOAD to commit after processing a specified number of rows. The default is 10000. OLLOAD prints an asterisk (*) to the screen each time it commits the specified number of rows. To disable the commit operation specify 0.

-mark *count*

Use this option if you want OLLOAD to print a dot on the screen after processing the specified number of records. The default is 1000. To disable this feature specify 0.

4.3.1.1 Comments

Data Parsing

Table 4–1 lists examples for OLLOAD data parsing.

Table 4–1 Data Parsing Examples

Input	Data	Explanation
'Redwood Shores, CA'	Redwood Shores CA	Enclosing the input string in quotes preserves spaces and punctuations within a string.
'O"Brien'	O'Brien	Represent a single quote with its escape sequence, two single quotes.
fire fly	firefly	Spaces in data that is not quoted is ignored.
,	NULL,NULL	Empty fields are NULL.
1,,3	1,NULL,3,NULL	Empty fields are NULL.
	[no row inserted]	Completely empty lines are ignored.

If there are more values than database columns, extra values are ignored. Any missing values at the end of the line are set to NULL.

OLLOAD Utility Restrictions

OLLOAD does not support tab-delimited input files and LONG datatypes.

Examples

```
olload -quote \" -file p_kakaku.csv -load c:\orant\oldb40\polite.odb skkm01

olload -dump c:\orant\oldb40\polite.odb emp empno ename | olload -load myfile.odb
myemp
```

5

Consolidator Performance (Consperf) Utility

This document provides a reference to the Consolidator Performance (Consperf) utility used to profile Consolidator publications. Each section of this document presents a different topic. Topics include:

- Section 5.1, "Overview"
- Section 5.2, "Generating Timing Statistics for Publications"
- Section 5.3, "Generating Explain Plans for Publications"
- Section 5.4, "Automatically Tuning Publication Properties"
- Section 5.5, "Analyzing Mobile Server Objects for Cost Based Optimizer"
- Section 5.6, "Usage Model"
- Section 5.7, "Consperf Parameters"

5.1 Overview

The Consperf utility is used to profile Consolidator publications. Application developers and administrators can use this utility to analyze performance of publications and identify potential bottlenecks during publication. This tool enables users to perform four primary functions:

- Generate Timing Statistics for Publications
- Generate Explain Plans for Publications
- Automatically Tune Publication Properties
- Analyze Mobile Server Objects for Cost Based Optimizer

During the Synchronization and MGP process, the Consolidator wraps publication item queries in templates to determine incremental changes. With complex snapshot queries, these templates can confuse the Oracle optimizer frequently and may result in poor execution plans. The Consperf utility exposes such templates and profiles their performance in conjunction with actual publication item queries. The Consperf utility generates SQL explain plans for each query.

Consperf Query Types

Query type codes are used in Consperf output files to identify various templates.

Table 5–1 lists Consperf query types used to identify templates and customized callouts.

Table 5–1 Consperf Query Types

Query Type	Description
BASE	Publication Item Query
NS	Null Sync Callout
BS	Before Sync Callout
SYNC_1	Sync template using outer-join for inserts, updates, and deletes
SYNC_2	Sync template for inserts and updates
SYNC_21	Sync templates for deletes
SYNC_22	Sync template for deletes using HASH_AJ
AS	After Sync Callout
BC	Before Compose Callout
LDEL_1	MGP template for logical deletes using EXISTS
LDEL_2	MGP template for logical deletes using correlated IN
LDEL_3	MGP template for logical deletes using HASH_AJ
LDEL_4	MGP template for logical deletes using IN
LINS_1	MGP template for logical inserts using EXISTS
LINS_2	MGP template for logical inserts using correlated IN
LINS_3	MGP template for logical inserts using IN
LG_1	MGP template for log updates using correlated IN
LG_2	MGP template for log updates using EXISTS
LG_3	MGP template for log updates using IN
LGN_1	MGP template for log updates with multiple table dependencies
AC	After compose callout
MAXLOG	MGP template to simulate log updates
DSYNC	Sync template to simulate records in the outbound client queue (used only by Consperf)
DMGP	MGP template to simulate records in the outbound client queue (used only by Consperf)
BASEVIEW	Indicates that a publication item can use an optimized base view. This template is generated by using the AUTOTUNE parameter and is referenced during synchronization.

5.2 Generating Timing Statistics for Publications

The Consperf utility enables application developers and administrators to generate timing statistics for publications. Users must specify the output file where the timing results are to be written. The timing statistics output is formatted into the following four distinct sections:

- Analyzing... This section displays parameters specified for generating timing statistics data and properties associated with the Mobile Server's installation.
- Sync Output... This section displays timing statistics for query types associated with the synchronization process. Based on the total time, publication items can be ordered from slowest to fastest.

- MGP Output... This section displays timing statistics for query types associated with the MGP process. Based on the total time, publication items can be ordered from slowest to fastest.
- Subscription Properties... This section displays subscription properties associated with each publication item.

Table 5–2 lists subscription properties and their corresponding description.

Table 5–2 Subscription Properties

Property	Description
Profiled	Indicates whether the publication item has been profiled with the Auto-Tune parameter enabled.
Base View	Indicates whether the publication item can use an optimized base view during synchronization. Both "Profiled" and "Base View" should be true before the base view is enabled.
Sub #	Indicates the number of records subscribed by the specified client.
Dirty #	Indicates the number of records currently marked as dirty for the specified client.
POPLOG	Indicates the number of records generated in the transaction log.
DSYNC	Indicates the number of records simulated in the outbound queue for the specified client in the Sync phase.
DMGP	Indicates the number of records simulated in the outbound queue for the specified client in the MGP phase.

Timing statistics for all publications are displayed in milliseconds. All the default query template types used during the Synchronization and MGP process are listed in brackets. For example, <10> represents a default query with a duration of 10 milliseconds.

5.3 Generating Explain Plans for Publications

Using the Consperf utility, application developers and administrators can generate explain plans. Users must specify an output file to store plan results. This output is divided by publication item and a plan is generated for all the relevant query templates.

5.4 Automatically Tuning Publication Properties

The Consperf utility is used to fine-tune a given publication's properties. Automatically tuned results generated for a profile can be used to complete a publication's properties. This function enables application developers and the administrator to choose the best performing query templates. In addition, the Consperf utility determines whether a publication item query can be replaced with an optimized base view. This replacement further improves synchronization performance when the publication item query is complex.

To tune publications, application developers and administrators must execute the Consperf utility with the `AUTOTUNE` parameter enabled.

For more information on running the Consperf utility with this option enabled, see Section 5.6, "Usage Model".

Table 5–3 displays Tuning results that are stored in the Mobile Server repository:

Table 5–3 C$CONSPERF TUNING RESULTS

Name	Null?	Type
PUBLICATION_ITEM	NOT NULL	VARCHAR(30)
QUERY TEMPLATE	NOT NULL	VARCHAR(10)

Configuring Query Templates

Using the `AUTOTUNE` feature of the Consperf utility, application developers and administrators can configure query templates.

Table 5–4 lists query templates that can be configured using the `AUTOTUNE` feature of the Consperf utility:

Table 5–4 Query Template Types

Query Template Type	Description
BASEVIEW	Use base view during synchronization
LDEL_1, LDEL_2, LDEL_3, LDEL_4	MGP templates for logical deletes
LINS_1, LINS_2, LINS_3	MGP templates for logical inserts
LG_1, LG_2, LG_3	MGP templates for log updates

5.5 Analyzing Mobile Server Objects for Cost Based Optimizer

The Consperf utility enables users to analyze Mobile Server objects so that the cost based optimizer can determine the appropriate query plans. Inaccurate statistics could be gathered because certain objects such as map tables and log tables may not contain any data. To counter this problem, the Consperf utility automatically simulates the appropriate data load before generating statistics.

5.6 Usage Model

Using the command line, the administrator and application developers can execute the Consperf utility. Users can call this utility as a win32 executable or as an executable java class. To call the Consperf utility, users must perform the following:

Win32 Executable

```
c:\consperf>consperf.exe [args...]
```

Java Class

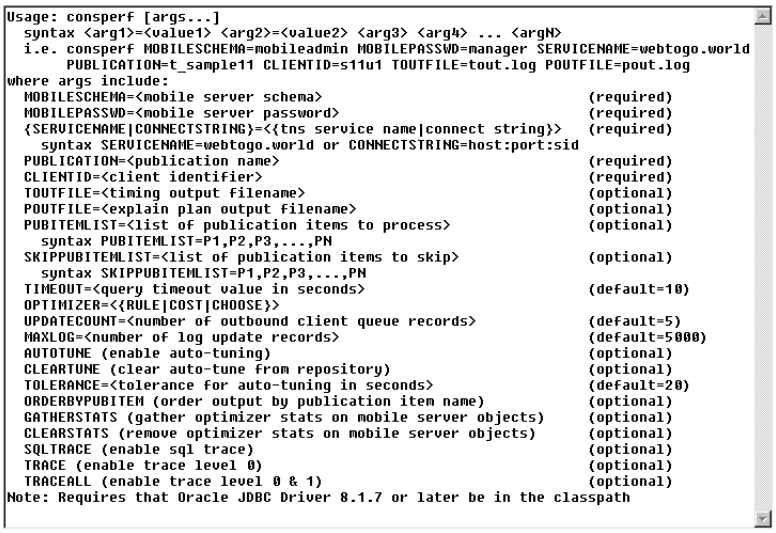
```
c:\consperf>java oracle.lite.sync.profiler.consperf [args...]
```

Two different types of arguments are passed to the Consperf utility. They are:

1. Assign: `<argN>=<valueN>` - specified argument with an associated value.
2. Switch: `<argN>` - specified argument turns on functionality.

Figure 5–1 displays a sample output after running the Consperf utility at the command line.

Figure 5–1 Sample Output Using Consperf at the Command Line



```
Usage: consperf [args...]
  syntax <arg1>=<value1> <arg2>=<value2> <arg3> <arg4> ... <argN>
  i.e. consperf MOBILESCHEMA=mobileadmin MOBILEPASSWD=manager SERVICENAME=webtogo.world
       PUBLICATION=t_sample11 CLIENTID=s11u1 TOUTFILE=tout.log POUTFILE=pout.log
where args include:
  MOBILESCHEMA=<mobile server schema>                              (required)
  MOBILEPASSWD=<mobile server password>                            (required)
  {SERVICENAME|CONNECTSTRING}=<{tns service name|connect string}>  (required)
    syntax SERVICENAME=webtogo.world or CONNECTSTRING=host:port:sid
  PUBLICATION=<publication name>                                   (required)
  CLIENTID=<client identifier>                                     (required)
  TOUTFILE=<timing output filename>                                (optional)
  POUTFILE=<explain plan output filename>                          (optional)
  PUBITEMLIST=<list of publication items to process>               (optional)
    syntax PUBITEMLIST=P1,P2,P3,...,PN
  SKIPPUBITEMLIST=<list of publication items to skip>              (optional)
    syntax SKIPPUBITEMLIST=P1,P2,P3,...,PN
  TIMEOUT=<query timeout value in seconds>                         (default=10)
  OPTIMIZER=<{RULE|COST|CHOOSE}>
  UPDATECOUNT=<number of outbound client queue records>            (default=5)
  MAXLOG=<number of log update records>                            (default=5000)
  AUTOTUNE (enable auto-tuning)                                    (optional)
  CLEARTUNE (clear auto-tune from repository)                      (optional)
  TOLERANCE=<tolerance for auto-tuning in seconds>                 (default=20)
  ORDERBYPUBITEM (order output by publication item name)           (optional)
  GATHERSTATS (gather optimizer stats on mobile server objects)    (optional)
  CLEARSTATS (remove optimizer stats on mobile server objects)     (optional)
  SQLTRACE (enable sql trace)                                      (optional)
  TRACE (enable trace level 0)                                     (optional)
  TRACEALL (enable trace level 0 & 1)                              (optional)
Note: Requires that Oracle JDBC Driver 8.1.7 or later be in the classpath
```

Note: The Consperf utility requires the Oracle JDBC Driver 8.1.7 or a later version to be available in the classpath. The classpath must also include the webtogo.jar file.

Analyzing and Using Consperf Results

As an application developer, you must use the Consperf utility to analyze every publication defined by you. Based on the information contained in the timing and explain plan output files, you must identify potential trouble areas, and make the necessary modifications to ensure acceptable performance levels of the publication. If acceptable performance cannot be achieved through the available query templates, you must implement a "Customized Compose" or "MyCompose" feature. This is an advanced feature, and enables you to implement your own application logic to determine server changes that should be downloaded to each client.

Troubleshooting Publication Performance Issues

As an administrator, you must use the Consperf utility to troubleshoot performance issues that may arise during day to day usage of the Consolidator. As the amount of data and the number of clients increases, unforeseen performance problems may arise. You can use the AUTOTUNE functionality to overcome some of these issues. However, if an acceptable query template is not available, it may be necessary for the application developer to implement a "Customized Compose" or "MyCompose" for publication items with poor performance levels. This is an advanced feature, and enables the application developer to implement his own application logic to determine server changes that should be downloaded to each client.

For more information on the Customized Compose or MyCompose feature, see the Mobile Server Synchronization Guide.

5.7 Consperf Parameters

The Consperf utility accepts many parameters. Some of these parameters are required while the others are optional.

Table 5–5 lists valid Consperf parameters and their corresponding descriptions:

Table 5–5 Consperf Parameters

Name	Type	Required	Default	UOM	Description	Sample format and Usage Notes
MOBILESCHEMA	Assign	Yes	N/A	N/A	Specifies the Mobile Server schema owner.	MOBILESCHEMA=mobileadmin
MOBILEPASSWD	Assign	Yes	N/A	N/A	Specifies the Mobile Server schema password.	MOBILEPASSWD=manager
SERVICENAME or CONNECTSTRING	Assign	Yes	N/A	N/A	TNS Service Name - this specifies the Oracle sql*net service name and uses the thick JDBC Driver or Connect String - this specifies the Oracle connect string and uses the thin JDBC Driver.	SERVICENAME=webtogo.world or CONNECTSTRING=host:1521:sid
PUBLICATION or APPLICATION	Assign	Yes	N/A	N/A	Specifies the name of the publication to be profiled. or Specifies the name of the application to be profiled.	PUBLICATION=t_sample11 or APPLICATION=orders
ITERATIONS	Assign	No	2	N/A	Specifies the number of times to execute template queries for average.	ITERATIONS=5
CLIENTID	Assign	Yes	N/A	N/A	Specifies the name of the client to be profiled.	CLIENTID=s11u1
TOUTFILE	Assign	No	<none>	N/A	Specifies the name of the output file for timing data. Must be specified for timing to be executed.	TOUTFILE=tout.log

Table 5–5 (Cont.) Consperf Parameters

Name	Type	Required	Default	UOM	Description	Sample format and Usage Notes
POUTFILE	Assign	No	<none>	N/A	Specifies the name of the output file for explain plan data. Must be specified for explain plans to be executed.	POUTFILE=pout.log
PUBITEMLIST	Assign	No	<all>	N/A	Specifies list of publication items to process. The default is all publication items in the publication.	PUBITEMLIST=pi1,pi2,pi3,…,piN
SKIPPUBITEMLIST	Assign	No	<none>	N/A	Specifies list of publication items to skip.	SKIPPUBITEMLIST=pi1,pi2,pi3,…,piN
TIMEOUT	Assign	No	10	Seconds	Specifies the query timeout value. This is the amount of time Consperf will wait before it cancels a query.	TIMEOUT=20
OPTIMIZER	Assign	No	db	CHOOSE RULE COST	Specifies the optimizer mode to use within Oracle.	OPTIMIZER=rule
UPDATECOUNT	Assign	No	5	# records	Specifies the number of records to mark as dirty during synchronization.	UPDATECOUNT=100
MAXLOG	Assign	No	5000	# records	Specifies the number of records to put in the log table. Simulates the transaction log.	MAXLOG=10000
AUTOTUNE	Switch	No	<off>	N/A	Enables auto-tune.	AUTOTUNE
CLEARTUNE	Switch	No	<off>	N/A	Clears existing auto-tune results.	AUTOTUNE

Table 5–5 (Cont.) Consperf Parameters

Name	Type	Required	Default	UOM	Description	Sample format and Usage Notes
TOLERANCE	Assign	No	20	Seconds	Specifies the auto-tune tolerance. A template must be faster by this amount before it replaces the default template.	TOLERANCE=60
ORDERBYPUBITEM	Switch	No	<off>	N/A	Orders all output by publication item name.	ORDERBYPUBITEM
GATHERSTATS	Switch	No	<off>	N/A	Gathers optimizer statistics on all mobile server objects.	GATHERSTATS MGP compose MUST be disabled while Consperf analyzes objects. Consperf blocks this automatically, but the safest approach is to manually stop the MGP process before running Consperf with the GATHERSTATS option. If Consperf fails while gathering statistics, users must re-run CLEARSTATS before starting the MGP process again.
CLEARSTATS	Switch	No	<off>	N/A	Removes optimizer statistics on mobile server objects.	CLEARSTATS
SQLTRACE	Switch	No	<off>	N/A	Enables Oracle sql trace. TKPROF can be used to analyze the resulting trace file.	SQLTRACE
TRACE	Switch	No	<off>	N/A	Enables trace level 0.	TRACE
TRACEALL	Switch	No	<off>	N/A	Enables trace level 0 and 1.	TRACE_ALL

Glossary

Apache Server

The Apache Server is a public domain HTTP server derived from the National Center for Supercomputing Applications (NCSA).

Base Table

A source of data, either a table or a view, that underlies a view. When you access data in a view, you are really accessing data from its base tables.

Connected

Connected is a generic term that refers to users, applications, or devices that are connected to a server. The Mobile Client for Web-to-Go is "connected" when it is in online mode.

Database Object

A database object is a named database structure: a table, view, sequence, index, snapshot, or synonym.

Database Server

The database server is the third tier of the Web-to-Go three-tier Web model. It stores the application data.

Disconnected

Disconnected is a generic term that refers to users, applications, or devices that are not connected to a server. The Mobile Client for Web-to-Go is "disconnected" when it is in offline mode.

Foreign Key

A foreign key is a column or group of columns in one table or view whose values provide a reference to the rows in another table or view. A foreign key generally contains a value that matches a primary key value in another table.

Index

An index is a database object that provides fast access to individual rows in a table. You create an index to accelerate the queries and sorting operations performed against the table's data. You also use indexes to enforce certain constraints on tables, such as unique and primary key constraints.

Indexes, once created, are automatically maintained and used for data access by the database engine whenever possible.

Integrity Constraint

An integrity constraint is a rule that restricts the values that can be entered into one or more columns of a table.

Java Applets

Java applets are small applications that are executed in the browser that extend the functionality of HTML pages by adding dynamic content.

Java Server Pages

JavaServer Pages (JSP) is a technology that enables developers to change a page's layout without altering the page's underlying content. JSP, which uses HTML and pieces of Java code to combine the presentation of dynamic content with business logic.

Java Servlets

Java servlets are protocol and platform-independent server-side components that are written in Java. Java servlets dynamically extend Java-enabled servers and provide a general framework for services built using the request-response paradigm.

Java Servlet Development Kit

The Java Servlet Development Kit is a tool provided by JavaSoft for developing Java servlets.

Java Web Server Development Kit

The Java Web Server Development Kit 1.0.1 is a JavaSoft tool for developing both JavaServer Pages (JSP) and Java servlets.

JDBC

JDBC (Java Database Connectivity) is a standard set of java classes providing vendor-independent access to relational data. Modeled on ODBC, the JDBC classes provide standard features such as simultaneous connections to several databases, transaction management, simple queries, manipulation of pre-compiled statements with bind variables, and calls to stored procedures. JDBC supports both static and dynamic SQL.

Join

A relationship established between keys (both primary and foreign) in two different tables or views. Joins are used to link tables that have been normalized to eliminate redundant data in a relational database. A common type of join links the primary key in one table to the foreign key in another table to establish a master-detail relationship. A join corresponds to a WHERE clause condition in a SQL statement.

Leapfrog Sequence

The leapfrog sequence is one of two sequence types that Web-to-Go uses in order to provide unique primary key values to the Mobile Client for Web-to-Go when it is in offline mode. Leapfrog sequences contain a different start value for each client, and each sequence increment is set to a larger value than the maximum number of clients.

Master Detail Relationship

A master-detail relationship exists between tables or views in a database when multiple rows in one table or view (the detail table or view) are associated with a single master row in another table or view (the master table or view).

Master and detail rows are normally joined by a primary key column in the master table or view that matches a foreign key column in the detail table or view.

When you change values for the primary key, the application should query a new set of detail records, so that values in the foreign key match values in the primary key. For example, if detail records in the EMP table are to be kept synchronized with master records in the DEPT table, the primary key in DEPT should be DEPTNO, and the foreign key in EMP should be DEPTNO. See also "Primary Key" and "Foreign Key".

MIME

MIME (Multipurpose Internet Mail Extensions) is a message format used on the Internet to describe the contents of a message. MIME is used by HTTP servers to describe the type of file being delivered.

MIME TYPE

MIME Type is a file format defined by Multipurpose Internet Mail Extension (MIME).

Mobile Client for Web-to-Go

The Mobile Client for Web-to-Go is the client tier of the Web-to-Go three-tier Web model. It contains the Mobile Client Web Server and the Oracle Database Lite database. Web-to-Go replicates the user's applications and data to Oracle Database Lite when the user switches to offline mode. When the user switches back to online mode, Web-to-Go replicates any data changes to the Oracle database.

Mobile Development Kit for Web-to-Go

The Mobile Development Kit for Web-to-Go enables application developers to develop and debug Web-to-Go applications that consist of Java servlets, JavaServer Pages (JSP), or Java applets.

Mobile Manager

The Mobile Manager is a web based application that runs in the browser for easy administration of Web-to-Go applications and users. Administrators use the Mobile Manager to perform functions such as granting or revoking application access to users or groups, modifying snapshot template variables, or deleting applications from Web-to-Go.

Mobile Server

The Mobile Server resides on the application server tier of the three-tier Web-to-Go model and processes requests from the Mobile Client for Web-to-Go to modify data in the database server. The Mobile Server can be configured to run with the Oracle HTTP Server, the Apache server, and the standalone Mobile Server.

Mobile Server Repository

The Mobile Server repository is a virtual file system that resides on an Oracle database. It is a persistent resource repository that contains all application files and definitions of the applications.

ODBC

ODBC (Open Database Connectivity) is a Microsoft standard that enables database access on different platforms. You can enable ODBC support on the Mobile Client for Web-to-Go for troubleshooting purposes. ODBC support enables you to view the client's data, which is stored on a local Oracle Database Lite database. To view this information, you can use SQL*Plus.

Oracle Database

The Oracle database is the database component of the Mobile Server. When the Mobile Client for Web-to-Go is in online mode, it stores applications and data on the Oracle database.

Oracle Database Lite

Oracle Database Lite is the database component of the Mobile Client for Web-to-Go. When the client is in offline mode, it stores applications and data on Oracle Database Lite.

Offline Mode

Offline mode is the condition of the Mobile Client for Web-to-Go when it is disconnected from the Mobile Server. In offline mode, the client applications are executed locally and data is accessed and stored in Oracle Database Lite. See also "Online Mode".

Online Mode

Online mode is the condition of the Mobile Client for Web-to-Go when it is connected to the Mobile Server. See also "Offline Mode".

Packaging Wizard

The Packaging Wizard enables administrators to publish Web-to-Go applications to the Mobile Server repository. Administrators can use the Packaging Wizard to create a new Web-to-Go application or to edit an existing application definition.

Primary Key

A table's primary key is a column or group of columns used to uniquely identify each row in the table. The primary key provides fast access to the table's records, and is frequently used as the basis of a join between two tables or views. Only one primary key may be defined per table.

Publication Item

To satisfy a PRIMARY KEY constraint, no primary key value can appear in more than one row of the table, and no column that is part of the primary key can contain a NULL value.

Referential Integrity

Referential integrity is defined as the accuracy of links between tables in a master-detail relationship that is maintained when records are added, modified, or deleted.

Carefully defined master-detail relationships promote referential integrity. Constraints in your database enforce referential integrity at the database (the server in a client/server environment).

The goal of referential integrity is to prevent the creation of an orphan record, which is a detail record that has no valid link to a master record. Rules that enforce referential integrity prevent the deletion or update of a master record, or the insertion or update of a detail record, that creates an orphan record.

Registry

The registry contains unique Web-to-Go name/value pairs. All registry names must be unique.

Replication

Replication is the process of copying and maintaining database objects in multiple databases that make up a distributed database system. Changes applied at one site are captured and stored locally before being forwarded and applied at each of the remote locations. Replication provides users with fast, local access to shared data, and protects the availability of applications because alternate data access options exist. Even if one site becomes unavailable, users can continue to query or even update the remaining locations.

Replication Conflict

Replication conflicts occur when contradictory changes to the same data are made. Web-to-Go avoids replication conflicts by using sequence values for disconnected clients.

Schema

A schema is a named collection of database objects, including tables, views, indexes, and sequences.

Sequence

A sequence is a schema object that generates sequential numbers. After creating a sequence, you can use it to generate unique sequence numbers for transaction processing. These unique integers can include primary key values. If a transaction generates a sequence number, the sequence is incremented immediately whether you commit or roll back the transaction. See also "Window Sequence" and "Leapfrog Sequence".

Sites

Web-to-Go creates a database for each user on the Mobile Client for Web-to-Go. This database is called a site. A client can contain multiple sites, but only one site per user. Users can have multiple sites on different clients.

Snapshots

Snapshots are copies of application data that Web-to-Go captures in realtime from the Oracle database and downloads to the client before it goes offline. A snapshot can be a copy of an entire database table, or a subset of rows from the table. The first time a user goes offline, Web-to-Go automatically creates the snapshots on the client machine. Each subsequent time that a user goes online or offline, Web-to-Go either refreshes the snapshots with the most recent data, or recreates them depending on the complexity of the snapshot.

SQL

SQL, or Structured Query Language, is a non-procedural database access language used by most relational database engines. Statements in SQL describe operations to be performed on sets of data. When a SQL statement is sent to a database, the database engine automatically generates a procedure to perform the specified tasks.

Switching Modes

Switching modes is the process the Mobile Client for Web-to-Go uses to go offline or to go back online. When the client switches to offline mode, it downloads all of the applications and data required to work offline on Oracle Database Lite. When the client switches back to online mode synchronizes data changes on Oracle Database Lite with the Oracle database.

Synchronization

Synchronization is the process Web-to-Go uses to replicate data between the Mobile Client for Web-to-Go and the Oracle database. Web-to-Go replicates the user's applications and data to Oracle Database Lite when the user switches to offline mode. When the user switches back to online mode, Web-to-Go replicates any data changes to the Oracle database.

Synonym

A synonym is an alternative name, or alias, for a table, view, sequence, snapshot, or another synonym.

Table

A table is a database object that stores data that is organized into rows and columns. In a well designed database, each table stores information about a single topic (such as company employees or customer addresses).

Three-Tier Web Model

The three-tier Web model is an Internet database configuration that contains a client, a middle tier, and a database server. Web-to-Go architecture follows the three-tier Web model.

Transaction

A set of changes made to selected data in a relational database. Transactions are usually executed with a SQL statement such as ADD, UPDATE, or DELETE. A transaction is complete when it is either committed (the changes are made permanent) or rolled back (the changes are discarded).

A transaction is frequently preceded by a query, which selects specific records from the database that you want to change.

Unique Key

A table's unique key is a column or group of columns that are unique in each row of a table. To satisfy a UNIQUE KEY constraint, no unique key value can appear in more than one row of the table. However, unlike the PRIMARY KEY constraint, a unique key made up of a single column can contain NULL values.

View

A view is a customized presentation of data selected from one or more tables (or other views). A view is like a "virtual table" that allows you to relate and combine data from multiple tables (called base tables) and views. A view is a kind of "stored query" because you can specify selection criteria for the data that the view displays.

Views, like tables, are organized into rows and columns. However, views contain no data themselves. Views allow you to treat multiple tables or views as one database object.

Web-to-Go

Oracle Web-to-Go is a framework for the creation and deployment of mobile, Web-based, database applications. Web-to-Go contains a three-tier database architecture consisting of the Mobile Client for Web-to-Go, the Mobile Server and the Oracle database. It is centrally managed from the server and Web-to-Go applications can be run when Web-to-Go connected to the server (online) or disconnected from the server (offline). When Web-to-Go is offline it caches data locally and synchronizes the data with the server when it goes back online.

Window Sequence

The window sequence is one of two sequences Web-to-Go uses in order to provide unique primary key values to the Mobile Client for Web-to-Go when it is in offline mode. The window sequence contains a unique range of values. The range of values never overlaps with those of other clients. When a client uses all the values in the range of its sequence, Web-to-Go recreates the sequence with a new, unique range of values.

Workspace

The Mobile Server Workspace is a Web page that provides users with access to Web-to-Go applications. Web-to-Go generates the Workspace in the user's browser after the user logs in to Web-to-Go. The Workspace displays icons, links, and descriptions of all applications that are available to the user. An application is available to the user after the administrator publishes it to the Web-to-Go system and grants access privileges to the user.

Index

C

Consolidator Performance Utility
  Analyzing Mobile Server Objects for Cost Based Optimizer, 5-4
  Automatically Tuning Publication Properties, 5-3
  Generating Explain Plans for Publications, 5-3
  Generating Timing Statistics for Publications, 5-2
  Overview, 5-1
  Usage Model, 5-4
    Analyzing and Using Consperf Results, 5-5
    Troubleshooting Publication Performance Issues, 5-5
Consperf
  Overview, 1-2
Consperf Parameters, 5-5
CREATEDB, 3-6
  Keywords and Parameters, 3-6
  Syntax, 3-6

D

Database Object, Glossary-1
Database Server, Glossary-1
Database Tools and Utilities
  Overview, 1-1
Database Tools and Utilities for PALM, 3-4
  Installing ValidateDB, 3-4
  Must Send Databases, 3-5
  Overview, 3-4
  Running ValidateDB, 3-4
  Sending Corrupted Databases, 3-5
DECRYPDB, 3-7
  Keywords and Parameters, 3-7
  Syntax, 3-7
Defining Application DDLs
  Importing Views and Index Definitions, 2-21
Defining Snapshots for Replication
  Creating Indexes for Snapshots, 2-13
  Creating New Snapshots, 2-12
  Editing Snapshots, 2-15
  Importing Sequences, 2-18
  Importing Snapshots, 2-14

E

ENCRYPDB, 3-8
  Keywords and Parameters, 3-8
  Syntax, 3-8

F

Foreign Key, Glossary-1

I

Integrity Constraint, Glossary-2

L

Leapfrog Sequence, Glossary-2
Load Utility
  Keywords and Parameters, 4-1
  Options, 4-2
  Overview, 4-1
  Syntax, 4-1

M

Master Detail Relationship, Glossary-2
MIGRATE, 3-10
  Keywords and Parameters, 3-10
  Syntax, 3-10
Mobile SQL, 3-2
  Database Access, 3-3
  SET TERM {ON|OFF}, 3-3
  SET TIMING {ON|OFF}, 3-3
  SET VERIFY {ON|OFF}, 3-3
  Starting Mobile SQL, 3-3
MSQL, 3-2
  MSQL for Windows 32, 3-2
  MSQL for Windows CE, 3-3
MSQL for Windows CE
  Populating your Database Using Mobile SQL, 3-2
  Using Mobile SQL, 3-3

O

ODBC Administrator and the Oracle Database Lite ODBC Driver, 3-11
  Adding a DSN Using the ODBC

Administrator, 3-13
Adding a DSN which points to Read-Only Media (CD-ROM), 3-13
ODBINFO, 3-13
Parameters, 3-14
Syntax, 3-13
OLLOAD, 3-15
Keywords and Parameters, 3-15
Syntax, 3-15

P

Packaging Wizard
Overview, 1-1
Publication Item, Glossary-4

R

Referential Integrity, Glossary-4
REMOVEDB, 3-18
Keywords and Parameters, 3-18
Syntax, 3-18
Replication Conflict, Glossary-5

S

Starting the Packaging Wizard
Web Applications, 2-2
Synchronization, Glossary-6
Synchronizing with an Encrypted Database, 3-10

U

Unique Key, Glossary-6
Using Mobile SQL
Database Access, 3-4
Starting Mobile SQL, 3-4
Using the Packaging Wizard
Web Applications, 2-1
Using the Packaging Wizard for Web Applications, 2-1
Adding Servlets, 2-7
Defining Application DDLs, 2-20
Defining Application Roles, 2-9
Defining Sequences for Replication, 2-16
Defining Snapshots for Replication, 2-10
Editing Applications, 2-21
Entering Database Information, 2-8
Listing Application Files, 2-5
Naming New Applications, 2-3
Starting the Packaging Wizard, 2-2, 2-22

V

VALIDATEDB, 3-18
Keywords and Parameters, 3-19
Syntax, 3-19