

Oracle® Database Lite

Administration and Deployment Guide

10g (10.0.0)

Part No. B12262-01

June 2004

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

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

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Send Us Your Comments	xi
Preface	xiii
Intended Audience.....	xiii
Documentation Accessibility	xiii
Structure	xiii
1 Introduction	
1.1 What is the Mobile Server.....	1-1
1.2 Concepts and Terminology	1-2
1.3 Implementation Tutorial.....	1-2
1.4 Detailed Instructions	1-2
1.5 Conventions.....	1-2
2 Getting Started	
2.1 The Overall Flow.....	2-1
3 Mobile Manager Overview	
3.1 Logging On	3-1
3.2 Mobile Manager Overview.....	3-2
3.2.1 Mobile Servers.....	3-3
3.2.1.1 Mobile Manager Home Page	3-4
3.2.1.2 Administering Applications	3-4
3.2.1.3 Administering Users	3-4
3.2.1.4 Mobile Server Administration	3-4
3.2.2 Mobile Devices	3-5
3.2.2.1 Devices	3-5
3.2.2.2 Platforms.....	3-6
3.2.2.3 Administration.....	3-7
4 Administering Mobile Applications	
4.1 Overview	4-1
4.2 Listing Applications	4-1
4.3 Modifying Application Properties	4-2

4.4	Suspending Mobile Applications	4-4
4.5	Resuming Mobile Applications	4-4
4.6	Uploading Applications to the Mobile Server Repository.....	4-4
4.6.1	Creating Database Tables	4-5
4.7	Removing Applications from the Mobile Server.....	4-5
4.8	Modifying Data Subsetting Parameters.....	4-5
4.9	Modifying Registry Entries	4-6
4.10	Selecting Application Files for Public Use	4-7
4.11	Adding Web Application Archive (WAR) Files.....	4-7
4.12	Adding Bookmarks.....	4-8
5	Administering Users	
5.1	Overview	5-1
5.2	Standalone Mode	5-1
5.2.1	Displaying User Groups and Individual Users.....	5-1
5.2.2	Adding New Groups.....	5-3
5.2.3	Adding New Users	5-4
6	Provisioning	
6.1	Managing Access Privileges.....	6-1
6.1.1	Granting or Revoking Application Access to Users and Groups	6-1
6.1.2	Including or Excluding Users from Group Based Access.....	6-2
6.2	Managing Snapshots	6-3
6.2.1	Read-only Snapshots.....	6-4
6.2.2	Updatable Snapshots.....	6-4
6.2.3	Refreshing a Snapshot.....	6-4
6.2.4	Snapshot Template Variables.....	6-4
7	Administering the Mobile Server	
7.1	Viewing System Status Reports for the Server	7-1
7.2	Suspending Applications.....	7-2
7.3	Resuming Applications.....	7-2
7.4	Viewing Active User Sessions	7-2
8	Managing Synchronization	
8.1	Overview	8-1
8.2	The Data Synchronization Manager: An Introduction.....	8-1
8.3	Managing the Sync Service.....	8-2
8.3.1	Starting/Stopping the Sync Service	8-3
8.3.2	Checking Sync Service Alerts.....	8-4
8.3.3	Managing Active Sync Sessions	8-5
8.3.4	Managing the Session History List	8-5
8.3.5	Displaying Operating System (OS) and Java Virtual Machine (JVM) Information... ..	8-6
8.4	Monitoring and Analyzing Sync Performance.....	8-7
8.4.1	Viewing Sync Statistics	8-7
8.4.2	Analyzing Performance of Publications.....	8-8

8.5	Administering the Synchronization Service	8-9
8.5.1	Managing Configuration Parameters	8-10
8.5.2	Managing Trace Settings and Trace Files	8-10
8.6	Browsing the Sync Repository	8-15
8.6.1	Viewing User Information.....	8-15
8.6.2	Viewing Publications	8-17
8.6.3	Viewing Publication Items	8-17
8.6.4	Viewing Publication Queues.....	8-18
8.7	Scheduling MGP Cycles to Run Inside the Mobile Server.....	8-19
8.8	Running the Message Generator and Processor (MGP) from the Command Line.....	8-20
8.8.1	Overview	8-21
8.8.2	MGP Cycles	8-21
8.8.3	Configuring MGP	8-21
8.9	Monitoring Synchronization Using SQL Scripts	8-22
8.9.1	Shared Maps	8-22
8.9.2	Synchronization Times for All Clients.....	8-23
8.9.3	Failed Transactions for all Clients	8-23
8.9.4	Completely Refreshed Publication Items for all Clients	8-23
8.9.5	Publications Flagged for Complete Refresh for All Clients.....	8-23
8.9.6	Clients and Publication where Subscription Parameters are Not Set	8-23
8.9.7	Record Counts for Map-based Publication Item by Client.....	8-24
8.9.8	Record Count for Map-based Publication Items by Store	8-24
8.9.9	All Client Sequence Partitions and Sequence Values	8-24
8.9.10	All Publication Item Indexes	8-24

9 Job Scheduler

9.1	Overview	9-1
9.2	Managing the Job Engine.....	9-2
9.2.1	Starting the Job Scheduler	9-2
9.2.1.1	Stopping the Job Scheduler	9-4
9.2.2	Checking Job Scheduler Alerts	9-4
9.2.3	Managing Active Jobs	9-4
9.2.4	Managing the Job History List.....	9-4
9.3	Administering Scheduled Jobs.....	9-5
9.3.1	Creating a New Job.....	9-5
9.3.2	Editing Existing Jobs	9-8
9.3.3	Enabling Jobs	9-9
9.3.4	Disabling Jobs.....	9-9
9.3.5	Deleting Jobs.....	9-9
9.3.6	Default Jobs.....	9-9
9.3.6.1	MGP_DEFAULT	9-10
9.3.6.2	PURGE_HISTORY_DEFAULT.....	9-10

10 Managing Devices

10.1	Overview and Architecture	10-1
10.1.1	Functionality.....	10-1

10.1.2	Terms and Definitions.....	10-2
10.1.3	Architecture	10-4
10.1.4	Security	10-5
10.1.4.1	Database Security	10-5
10.1.4.2	Communication Security	10-5
10.1.4.3	Command Execution.....	10-5
10.1.4.4	Database Synchronization Security	10-6
10.2	Device Management Client	10-6
10.2.1	Device Management Agent.....	10-6
10.2.1.1	Bootstrap Process.....	10-7
10.2.1.2	Configuration Parameters	10-7
10.2.1.3	User Interface	10-8
10.2.2	Software Update Client.....	10-8
10.2.3	Software Version Checker	10-9
10.3	Software Management Client (SMC)	10-9
10.3.1	Software Installation.....	10-9
10.3.2	Installation Configuration (INF) File	10-9
10.3.2.1	Setup Information.....	10-10
10.3.2.2	Properties.....	10-10
10.3.2.3	Initialization	10-11
10.3.2.4	Including Other INF Files.....	10-11
10.3.2.5	INSTALL Element	10-11
10.4	Device Management Server.....	10-14
10.4.1	Overview	10-15
10.4.2	Mobile Manager	10-15
10.4.3	Managing Device Platforms.....	10-15
10.4.3.1	Modifying Platform Properties.....	10-16
10.4.3.2	Creating Custom Platform	10-16
10.4.4	Managing Device Commands.....	10-16
10.4.4.1	Modifying Existing Commands	10-17
10.4.4.2	Creating New Commands.....	10-17
10.4.4.3	Creating Group Commands.....	10-17
10.4.5	Managing Device Configuration	10-17
10.4.5.1	Modifying Client Configuration Items.....	10-18
10.4.5.2	Creating Configuration Items	10-18
10.4.6	Managing Network Providers	10-18
10.4.6.1	Modifying Network Provider Properties.....	10-19
10.4.6.2	Creating New Network Providers	10-19
10.4.7	Administering Devices	10-19
10.4.7.1	Modifying Device Properties.....	10-19
10.4.7.2	Viewing Applications	10-20
10.4.7.3	Viewing Device Information.....	10-20
10.4.7.4	Scheduling or Sending Commands	10-20
10.4.7.5	Viewing Device Logs	10-20
10.4.7.6	Viewing the Device Command History	10-21

11	Tracing	
11.1	Overview	11-1
11.2	Setting Trace Parameters	11-2
11.2.1	Mobile Server Modes	11-3
11.3	Editing the webtogo.ora File	11-3
11.4	Running the Mobile Server in Debug Mode	11-4
12	Deployment	
12.1	Overview	12-1
12.2	How To Deploy	12-1
13	Deployment: Creating an Install CD for Mobile Clients	
13.1	How to Create an Offline CD for Mobile Clients	13-1
13.1.1	Downloading the Offline Install Files.....	13-1
13.1.2	Compile the Offline Install Files.....	13-1
13.2	Installing a Mobile Client from the CD.....	13-2
14	Deployment: Using the Mobile Client for Web-to-Go	
14.1	Overview	14-1
14.2	Installing the Mobile Client for Web-to-Go	14-1
14.2.1	Hardware Requirements	14-1
14.2.2	Software Requirements.....	14-2
14.2.3	Installation	14-2
14.3	Client-Side Configuration for Secure Socket Layer (SSL)	14-3
14.3.1	Communication between the Mobile Client and the Mobile Server.....	14-3
14.3.2	Connection between the Browser and the Mobile Client for Web-to-Go.....	14-4
14.4	Support for Non-SSL Mobile Clients	14-4
15	Administering and Using the Branch Office	
15.1	Introduction	15-1
15.1.1	What is the Branch Office?	15-1
15.1.2	How the Branch Office Works	15-2
15.1.3	The Branch Office Manager.....	15-3
15.1.4	Synchronizing Data with Headquarters.....	15-3
15.2	Branch Office Installation and Configuration.....	15-3
15.2.1	Terms and Concepts.....	15-3
15.2.2	Overview.....	15-4
15.2.3	Branch Office Installation	15-4
15.3	Architecture	15-7
15.3.1	The Branch Office Environment.....	15-7
15.3.1.1	The Branch Office Client.....	15-7
15.3.1.2	The Branch Office	15-8
15.3.1.3	Company Headquarters	15-9
15.3.2	Connecting Clients to the Branch Office Database Machine.....	15-9
15.3.2.1	ODBC Connection	15-9

15.3.2.2	JDBC Connections	15-9
15.4	Administration	15-10
15.4.1	Overview	15-10
15.4.1.1	Database Service Administration.....	15-10
15.4.1.2	Remote User Maintenance	15-11
15.4.2	Logging into the Branch Office Manager	15-11
15.4.3	Using the Branch Office Manager	15-11
15.4.3.1	Updating Status Summary	15-12
15.4.3.2	Starting the Database Service.....	15-12
15.4.3.3	Stopping the Database Service	15-12
15.4.3.4	Viewing the Status of the Branch Office Database	15-12
15.4.4	Managing Branch Office Users	15-13
15.4.4.1	Creating Users.....	15-13
15.4.4.2	Setting User Roles.....	15-13
15.4.4.3	Setting User Properties	15-14
15.4.4.4	Setting User Privileges	15-14
15.4.4.5	Finding Users	15-14
15.4.4.6	Removing a User	15-14
15.4.5	Managing Applications.....	15-14
15.4.5.1	Installing Public Files	15-15

16 Offline Instantiation

16.1	Overview	16-1
16.2	Setting Up the Mobile Server Host and Mobile Development Kit Host.....	16-1
16.3	Setting Up the Mobile Client Root Directory.....	16-1
16.4	Setting Up the Mobile Client (Distribution) Root Directory	16-2
16.5	The OLI CD's Directory	16-2
16.6	The OLI Configuration File: Oli.ini	16-3
16.7	Using the OLI Engine	16-5
16.7.1	Using Sub-Commands	16-5
16.7.1.1	makeodb.....	16-6
16.7.1.2	package.....	16-6
16.7.1.3	cleanup	16-6
16.8	Deploying on Client Machines.....	16-6

17 Configuring SSL For Mobile Server

17.1	Creating an SSL Certificate.....	17-1
17.2	Configuring Mobile Server for SSL	17-2
17.2.1	Configuring SSL for Mobile Server With Oracle9iAS or OracleAS.....	17-2
17.2.2	Configuring SSL for Standalone Mobile Server	17-3
17.3	Troubleshooting Error Messages for an SSL-Enabled Mobile Server	17-4

A Troubleshooting FAQ's

A.1	Inspecting Files in the Mobile Server Repository.....	A-1
A.2	Running the Mobile Server in Debug Mode	A-2
A.3	First Synchronization Causes Browser to Timeout.....	A-3

A.4	Accessing the Client Database Offline	A-3
A.5	Accessing Schema Changes to a Published Application.....	A-4
B Mobile Server Configuration Parameters		
B.1	[WEBTOGO]	B-1
B.2	[FILESYSTEM].....	B-4
B.3	[DEBUG].....	B-5
B.4	[PUBLIC]	B-8
B.5	[SERVLET_PARAMETERS]	B-8
B.6	[CONSOLIDATOR].....	B-9
C Consolidator Requirements in INIT.ORA		
C.1	Relationships Between Relevant Parameters.....	C-1
C.2	Values for Processes and DML Locks	C-1
D Scripting Language for the Mobile Server		
D.1	Description of the Syntax.....	D-1
D.2	Running a Script INI File	D-5
D.3	Examples	D-5
D.3.1	Creating, Adding, and Granting Access	D-5
D.3.2	Deleting, Removing, and Revoking Access	D-7
E External Authentication		
E.1	Using External Authentication.....	E-1
E.2	External Authentication Code Sample.....	E-2
F Bypassing a Proxy Server		
G System Catalog Views		
G.1	Mobile Server Catalog Views.....	G-1
G.1.1	"MOBILEADMIN".CV\$ALL_CLIENTS	G-1
G.1.2	"MOBILEADMIN".CV\$ALL_ERROR.....	G-2
G.1.3	"MOBILEADMIN".CV\$ALL_PUBLICATIONS	G-2
G.1.4	"MOBILEADMIN".CV\$ALL_SUBSCRIPTIONS.....	G-2
G.1.5	"MOBILEADMIN".CV\$ALL_SEQUENCES	G-2
G.1.6	"MOBILEADMIN".CV\$ALL_SEQUENCE_PARTITIONS.....	G-3
G.1.7	"MOBILEADMIN".CV\$ALL_PUBLICATION_ITEMS_ADDED	G-3
G.1.8	"MOBILEADMIN".CV\$ALL_PUBLICATION_ITEMS	G-3
G.1.9	"MOBILEADMIN.CV\$ALL_PUBLICATION_ITEM_INDEXES	G-4
G.1.10	".....MOBILEADMIN"CV.\$ALL_SUBSCRIPTION_PARAMS	G-4
H Using the JDBC Thin Driver		
H.1	Upload the Oracle JDBC Driver.....	H-1
H.2	Change the Applet Tag	H-1

I POLITE.INI Database Parameters

I.1	POLITE.INI File Overview	I-1
I.2	POLITE.INI Parameters	I-1
I.2.1	CacheSize	I-1
I.2.2	Database ID.....	I-1
I.2.3	DBCharEncoding	I-1
I.2.4	MAXINDEXCOLUMNS	I-2
I.2.5	NLS_DATE_FORMAT	I-2
I.2.5.1	Date Format	I-2
I.2.5.2	Date Format Examples.....	I-3
I.2.6	NLS_LOCALE	I-4
I.2.7	NLS_SORT	I-5
I.2.8	ReverseJoinOrder.....	I-6
I.2.9	SharedAddress	I-6
I.2.10	SuggestedSharedAddress.....	I-6
I.2.11	SQL Compatibility	I-7
I.2.12	TempDB	I-7
I.2.13	TempDir	I-7
I.2.14	OLITE_SQL_TRACE	I-7
I.2.15	FLUSH_AFTER_WRITE	I-8
I.3	Sample POLITE.INI File.....	I-8

J The Consolidator Client API (OCAPI)

J.1	Overview	J-1
J.2	Sync Client Parameters	J-1
J.2.1	TIME_LOG.....	J-2
J.2.2	UPDATE_LOG	J-2
J.2.3	COMPRESSION	J-2
J.2.4	DEBUG	J-3
J.2.5	AUTO_COMMIT_COUNT	J-3
J.2.6	TEMP_DIR.....	J-3
J.2.7	RESUME_CLIENT_TIMEOUT	J-3
J.2.8	RESUME_CLIENT_MAXSEND	J-3
J.2.9	ERROR_REPORT	J-4
J.2.10	DB_ENCODING	J-4
J.2.11	MEM_THRESHOLD	J-4
J.2.12	VALIDATEDB.....	J-4
J.2.13	ENCRYPDB	J-5

Glossary

Index

Send Us Your Comments

Oracle Database Lite Administration and Deployment Guide 10g (10.0.0)

Part No. B12262-01

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 10p2
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 Administration and Deployment Guide*, discussing the intended audience, documentation accessibility, and structure of 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.

Accessibility of Links to External Web Sites in Documentation

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

Structure

This guide includes the following topics:

- [Chapter 1, "Introduction"](#)

This chapter describes the process of publishing, managing, and deploying your mobile applications using Oracle Database Lite 10g.

- [Chapter 2, "Getting Started"](#)

The information in this chapter enables you, the Administrator, to started publishing your applications into the Mobile Server Repository.

- [Chapter 3, "Mobile Manager Overview"](#)

This chapter provides an overview of the Mobile Manager. The Mobile Manager is a web based application and enables the Mobile Server Administrator to administer mobile applications.

- [Chapter 4, "Administering Mobile Applications"](#)

This chapter discusses how to administer mobile applications.

- [Chapter 5, "Administering Users"](#)

This chapter describes how to administer users and mobile applications using the Mobile Manager.

- [Chapter 6, "Provisioning"](#)

This chapter enables the Mobile Server Administrator to grant access privileges to mobile application developers and manage snapshots.

- [Chapter 7, "Administering the Mobile Server"](#)

This chapter enables users to administer the Mobile Server.

- [Chapter 8, "Managing Synchronization"](#)

This chapter enables the Mobile Server administrator to use the Data Synchronization Manager and manage synchronization tasks.

- [Chapter 9, "Job Scheduler"](#)

This chapter enables the Mobile Server administrator to manage the job engine and job schedules through the web.

- [Chapter 10, "Managing Devices"](#)

This chapter describes how to manage devices.

- [Chapter 11, "Tracing"](#)

This chapter enables the Mobile Server Administrator to set tracing parameters.

- [Chapter 12, "Deployment"](#)

This chapter enables the administrator to deploy Oracle Database Lite 10g.

- [Chapter 13, "Deployment: Creating an Install CD for Mobile Clients"](#)

This chapter describes how to create an install CD for installing Mobile Clients offline.

- [Chapter 14, "Deployment: Using the Mobile Client for Web-to-Go"](#)

This chapter enables end users of mobile applications to install and use the Mobile Client for Web-to-Go.

- [Chapter 15, "Administering and Using the Branch Office"](#)

This chapter describes how to install, configure, administer and use the Mobile Client for Branch Offices.

- [Chapter 16, "Offline Instantiation"](#)

This chapter discusses the Offline Instantiation feature.

- [Chapter 17, "Configuring SSL For Mobile Server"](#)

Oracle Database Lite 10g supports Secure Socket Layer (SSL) communication between the Mobile Server and Mobile Clients. Oracle Database Lite uses the SSL that is embedded within OC4J, which is shipped as part of Mobile Server.

- [Appendix A, "Troubleshooting FAQ's"](#)

This appendix contains frequently asked questions for troubleshooting the Mobile Server.

- [Appendix B, "Mobile Server Configuration Parameters"](#)

This appendix describes configuration parameters for the Mobile Server. These parameters are included in the file `webtogo.ora`. The Mobile Server uses the `webtogo.ora` file to initialize the Mobile Server.

- [Appendix C, "Consolidator Requirements in INIT.ORA"](#)

This appendix describes the Consolidator requirements for Oracle and Oracle parameter settings in the file `init.ora`.

- [Appendix D, "Scripting Language for the Mobile Server"](#)

This appendix describes the scripting language for the Mobile Server. You can use scripting to perform batch processing tasks that are performed frequently by the administrator.

- [Appendix E, "External Authentication"](#)

This appendix describes how to use external authentication mechanisms.

- [Appendix F, "Bypassing a Proxy Server"](#)

This appendix enables you to connect to the Mobile Server through a proxy server.

- [Appendix G, "System Catalog Views"](#)

This appendix is a reference to system catalog views for the Mobile Admin schema.

- [Appendix H, "Using the JDBC Thin Driver"](#)

This appendix enables you to use the JDBC thin driver with Mobile Server applets.

- [Appendix I, "POLITE.INI Database Parameters"](#)

This document discusses the `POLITE.INI` file and its associated parameters.

- [Appendix J, "The Consolidator Client API \(OCAPI\)"](#)

This appendix provides a reference to the Consolidator Client API parameters, enabling you, the Administrator, to modify the settings of each parameter.

Introduction

This chapter describes the process of publishing, managing, and deploying your mobile applications using Oracle Database Lite 10g. The primary audiences for this guide are the System or Database Administrator and the end user. After the application developer packages mobile applications, the Administrator publishes and manages mobile applications. At this stage, end users can begin deployment of Oracle Database Lite 10g applications.

For detailed information about deployment, see [Chapter 12, "Deployment"](#) and [Chapter 14, "Deployment: Using the Mobile Client for Web-to-Go"](#).

This document introduces the Mobile Server. The following topics are discussed.

- [Section 1.1, "What is the Mobile Server"](#)
- [Section 1.2, "Concepts and Terminology"](#)
- [Section 1.3, "Implementation Tutorial"](#)
- [Section 1.4, "Detailed Instructions"](#)
- [Section 1.5, "Conventions"](#)

1.1 What is the Mobile Server

The Mobile Server enables the Administrator to publish, manage, and provide mobile applications to a wide range of different device platforms. The Mobile Server performs both data and application synchronization to Mobile Clients.

Mobile application developers can use the Packaging Wizard to package (or define) the mobile application, specifying the snapshot definition information. This process includes describing your application, listing files that are part of the application, and the snapshots (publication items) that your mobile application needs. You can then save the application into a jar file. These tasks are typically executed by the developer of the mobile application. Once an application is packaged, the Administrator can publish the application into the Mobile Server Repository using the Mobile Manager. For a detailed description of the Packaging Wizard, refer the *Oracle Database Lite Tools and Utilities Guide*.

When the Administrator publishes an application, replication objects are automatically created in the Mobile Server Repository. A publication is created for each platform that is supported by the mobile application. The snapshot definition information is part of the application definition, which is specified by the developer and is stored in a "jar" file.

1.2 Concepts and Terminology

Before you implement Oracle Database Lite 10g, you should understand Oracle Database Lite 10g concepts and terminology. The *Oracle Database Lite Developer's Guide* provides a conceptual overview of Oracle Database Lite 10g. The [Glossary](#) chapter in this guide contains a complete list of terms and definitions. You should review these sections before getting started.

1.3 Implementation Tutorial

Once you understand Oracle Database Lite 10g concepts and terminology, you should become familiar with the implementation process. You can find tutorials on the implementation process that use the corresponding implementation platforms. For more information, refer the *Oracle Database Lite Developer's Guide*. The tutorials provide in this document guide you through the process of developing and running a sample applications for the chosen platform.

1.4 Detailed Instructions

Once you have developed an overview of developing and implementing Oracle Database Lite 10g applications for the chosen platform, you can follow the instructions provided in Chapter 4, "Developing Mobile Web Applications," and Chapter 5, "Native Application Development," in the *Oracle Database Lite Developer's Guide*. These chapters provide detailed instructions for developing and packaging your mobile applications. This guide also provides sample applications and troubleshooting information.

1.5 Conventions

The following conventions are used in this guide to clarify who performs a particular task when the topic being discussed includes a description of certain tasks:

- When the context of the discussion focuses on the Administrator's role, the text refers to the system administrator as the "Administrator" (the initial letter in uppercase). At this point, the Administrator, as the intended audience, is not addressed as "you" because it may not be clear to the reader, yet, who the intended audience is.
- When it becomes established in the context that it is the Administrator who performs the task, the text refers to the Administrator in the second person (that is, as an expressed or implied "you"). For example, "When you suspend the Mobile Server, the Mobile Server terminates all currently active sessions on the server....To perform this task, click the Server tab and then...."

Getting Started

The information in this chapter enables you, the Administrator, to start publishing your applications into the Mobile Server Repository. Before initiating the data synchronization process, you must first publish your mobile applications.

2.1 The Overall Flow

The general flow for publishing an application, whether a web based application or a native one, is as follows.

1. Ensure that the Mobile Server is running. Package the application using the Packaging Wizard. For a step-by-step description of how to use the Packaging Wizard, consult the *Oracle Database Lite Developer's Guide*.
2. If the application tables do not already exist in the Oracle database, you must create the requisite tables. For more information, see [Chapter 4, "Administering Mobile Applications"](#), Section 4.6.1, "Creating Database Tables".
3. Logon to the Mobile Manager and publish the application. For more information, see [Chapter 4, "Administering Mobile Applications"](#), Section 4.6, "Uploading Applications to the Mobile Server Repository".
4. Go to the Application Properties page. Set the database password for the application tables. For more information, see [Chapter 4, "Administering Mobile Applications"](#), Section 4.3, "Modifying Application Properties".
5. Go to the Users page and assign this application to users. For more information, see [Chapter 4, "Administering Mobile Applications"](#).
6. Navigate to the Data Subsetting page and set the data subsetting parameters, (if applicable) for each user on each platform. For more information, see [Chapter 4, "Administering Mobile Applications"](#), Section 4.8, "Modifying Data Subsetting Parameters".
7. Start the Message Generator and Processor (MGP). For more information, see [Chapter 8, "Managing Synchronization"](#), Section 8.7, "Scheduling MGP Cycles to Run Inside the Mobile Server".

Mobile Manager Overview

This chapter provides an overview of the Mobile Manager. The Mobile Manager is a web based application and enables the Mobile Server Administrator to administer mobile applications. Topics include:

- [Section 3.1, "Logging On"](#)
- [Section 3.2, "Mobile Manager Overview"](#)

3.1 Logging On

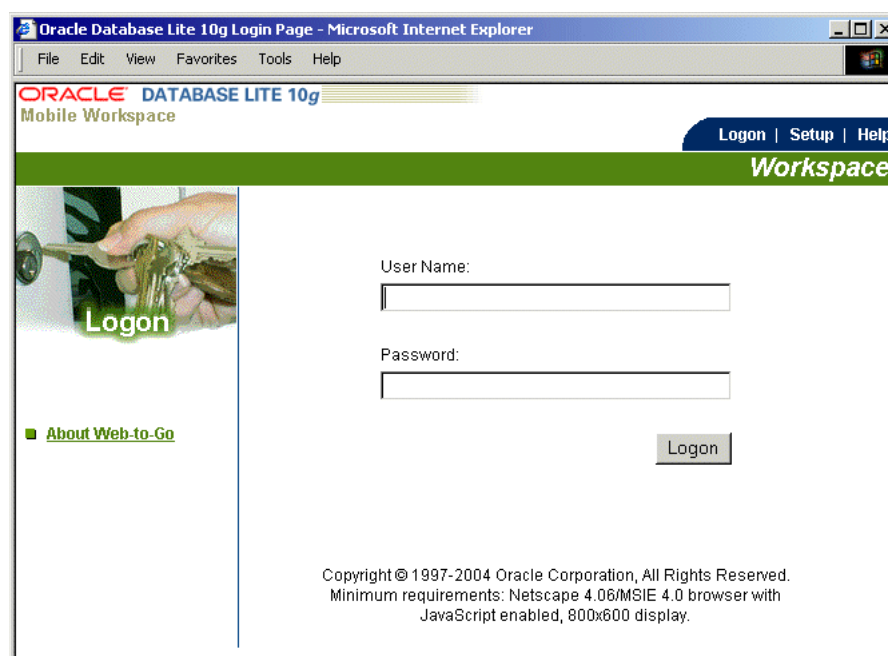
To logon to the Mobile Manager, perform the following steps.

1. Using a browser, connect to the Mobile Server by entering the following URL.

`http://<your_Mobile_Server_host_name>/webtogo`

As [Figure 3-1](#) displays, the Mobile Server logon page appears.

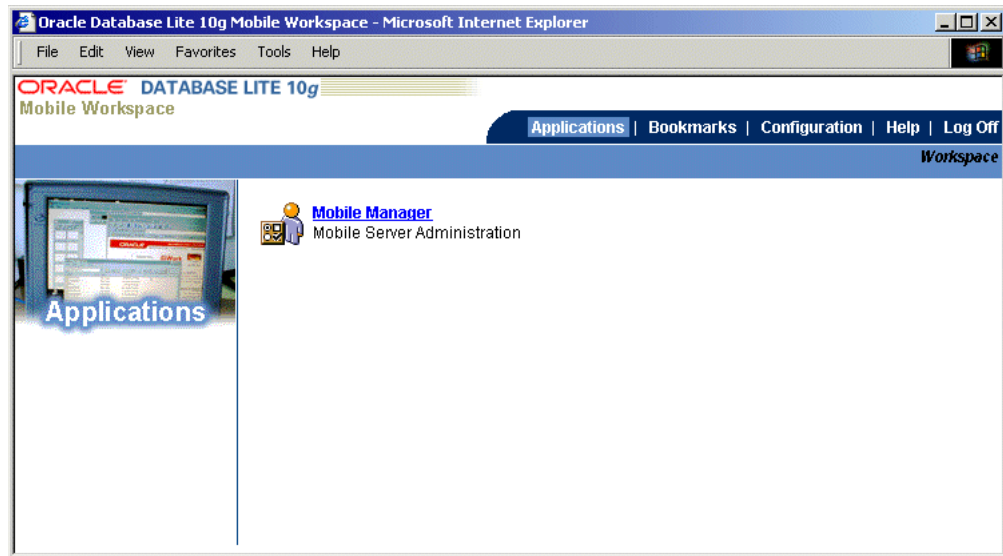
Figure 3-1 Logon Page



2. Log on to the Mobile Manager using "administrator" as the **User Name** and "admin" as the **Password**. The Mobile Manager displays the Workspace page.

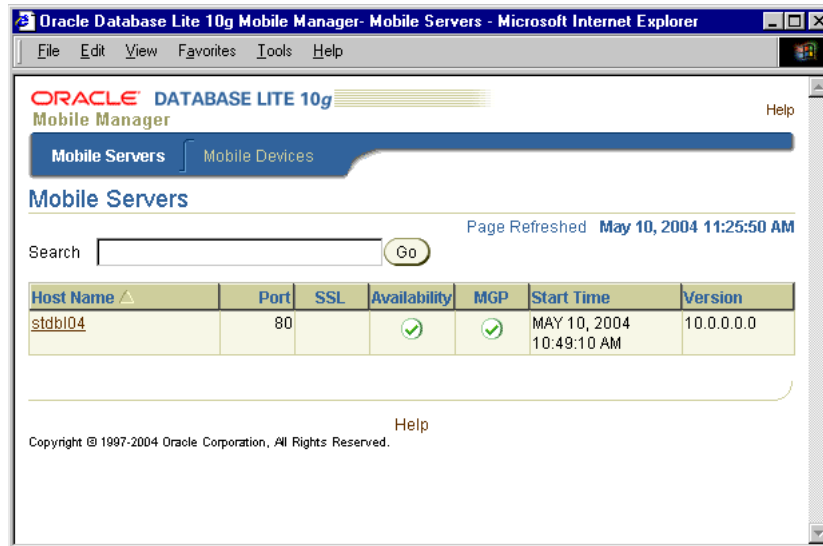
Figure 3–2 displays the Mobile Manager Workspace page.

Figure 3–2 Mobile Manager Workspace Page



3. Click the **Mobile Manager** icon or link. As Figure 3–3 displays, the Mobile Server Farms page appears.

Figure 3–3 Mobile Server Farms Page



3.2 Mobile Manager Overview

A Mobile Server Farm is a group of Mobile Servers configured to run against the same repository. Information pertinent to Mobile Server Farms is stored in a table in the Mobile Server Repository.

Your Mobile Server is registered on the Mobile Server Farms page during installation. To register the current instance of a Mobile Server, the following information is inserted into the repository.

- Mobile Server Instance ID
- Hostname
- Protocol
- IP Address
- Port
- Status
- Start Time
- Version
- Install Date

The Mobile Server Farms page contains Mobile Server and Mobile Devices tabs. The following sections describe each component of the Mobile Server Farm. Topics include:

- [Section 3.2.1, "Mobile Servers"](#)
- [Section 3.2.2, "Mobile Devices"](#)

3.2.1 Mobile Servers

The Mobile Servers tab lists all Mobile Servers that are configured to run against the same repository. As [Figure 3-4](#) displays, this page lists Mobile Server information such as Host name, Port, SSL enabled, Up or Down Status, the MGP instance, start time of the instance, and Mobile Server version.

Figure 3-4 Mobile Servers List

Oracle Database Lite 10g Mobile Manager - Mobile Servers - Microsoft Internet Explorer

ORACLE DATABASE LITE 10g Mobile Manager

Mobile Servers Mobile Devices

Mobile Servers

Page Refreshed May 26, 2004 11:24:05 PM

Search Go

Host Name	Port	SSL	Status	MGP	Start Time	Version
vkanchar-pc2	80		✓	✓	May 26, 2004 11:15:40 PM	10.0.0.0.0
bragupat-pc1	80		!		May 21, 2004 4:32:50 PM	10.0.0.0.0

Help

Copyright © 1997-2004, Oracle. All rights reserved.

If a Mobile Server instance is not running (status column displays down), the hyperlink for such host names is not enabled. If this page is refreshed after a Mobile Server instance is started or stopped (by clicking on Mobile Server link), the Mobile

Server's status is reflected on this page. Using search criteria based on host name, you can filter the Mobile Server display list.

The Mobile Server provides mobile application management and configuration functions. It groups existing mobile servers in a Mobile Server farm and manages mobile applications developed for every mobile server through the utility known as the Mobile Manager. The following sections describe the Mobile Manager's components.

- [Section 3.2.1.1, "Mobile Manager Home Page"](#)
- [Section 3.2.1.2, "Administering Applications"](#)
- [Section 3.2.1.3, "Administering Users"](#)
- [Section 3.2.1.4, "Mobile Server Administration"](#)

3.2.1.1 Mobile Manager Home Page

Your Mobile Manager home page maintains the following Mobile Server information.

- General information such as current Mobile Server status, version, and mode.
- Database information such as database version, JDBC URL, JDBC Driver, JDBC version, and schema name.
- Data Synchronization information such as MGP Status, In Queue, Out Queue, and Error Queue details.
- Alert details that describe alert severity and the date and time on which the alert was triggered.
- Mobile Server components such as Data Synchronization and the Job Scheduler that enable you to schedule synchronization job sessions.

3.2.1.2 Administering Applications

The Applications page enables the Mobile Server administrator to accomplish the following tasks.

1. Publish applications.
2. Create or edit application properties.
3. Resume, suspend, and delete applications.
4. Grant or revoke application access to users and groups.
5. Create or edit data subsetting parameters.
6. When required, provision mobile application files for public use.
7. Add WAR files.

3.2.1.3 Administering Users

The Users page enables the Mobile Server administrator to accomplish the following tasks.

1. Add groups.
2. Add users.

3.2.1.4 Mobile Server Administration

The Administration page enables the Mobile Server administrator to accomplish the following tasks.

1. View sessions.
2. Edit trace settings.
3. Edit the configuration file.
4. Add bookmarks.
5. View a summary of the database, JRE, and Operating System.

3.2.2 Mobile Devices

The Mobile Devices tab lists all mobile devices that are registered with any Mobile Server, and are part of the same Mobile Server farm. The following sections describe functionality in the Mobile Devices tab and includes the following topics.

- [Section 3.2.2.1, "Devices"](#)
- [Section 3.2.2.2, "Platforms"](#)
- [Section 3.2.2.3, "Administration"](#)

3.2.2.1 Devices

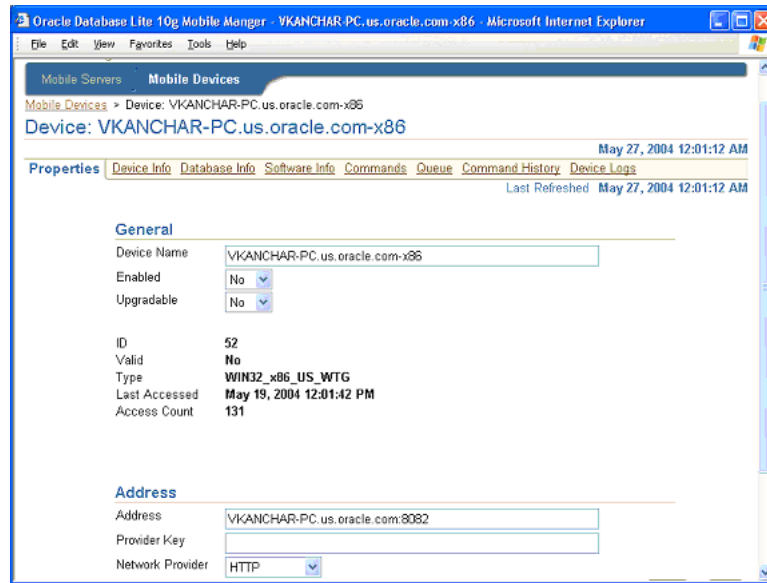
This page lists mobile device information such as device name, owner, platform, version, and date and time on which it was last accessed. [Figure 3–5](#) displays the Devices page.

Figure 3–5 *Devices Page*



To access the Device properties, users can click the Device Name link from the given device list. [Figure 3–6](#) displays the Device Properties page.

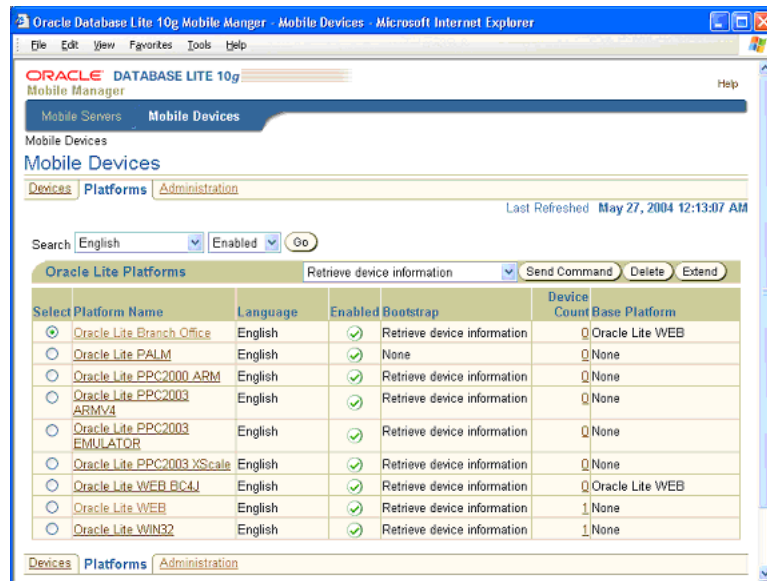
Figure 3–6 Device Properties Page



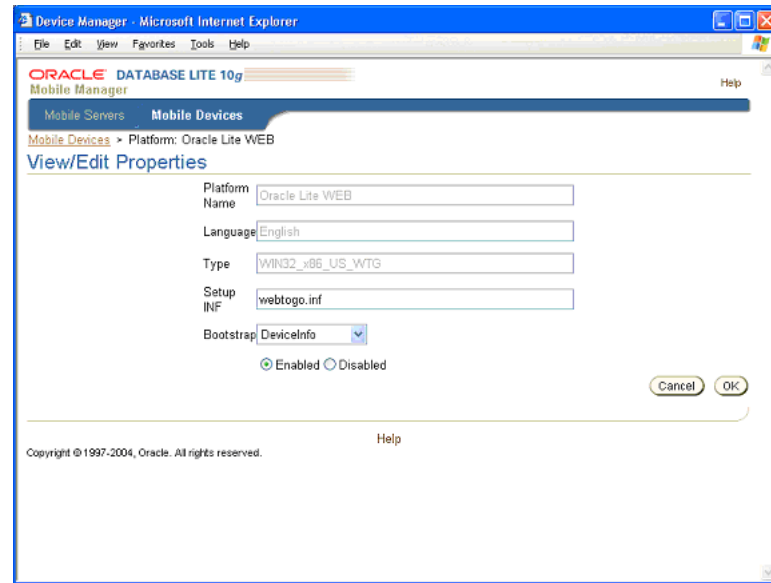
3.2.2.2 Platforms

This page lists platform information such as platform name, language, enabled, bootstrap, device count, and base platform. [Figure 3–7](#) displays the Platforms page.

Figure 3–7 Platforms Page



To view platform properties, users can click the required platform name link. [Figure 3–8](#) displays the Platform Properties page.

Figure 3–8 Platform Properties Page

3.2.2.3 Administration

This page lists the device administration options namely Command Management, Configuration Management, and Network Management. The Command Management page is used to manage Device commands. The Configuration Management can be used to change server side configuration and the Network Management page is used to change network related settings. [Figure 3–9](#) displays the Administration page.

Figure 3–9 Administration Page

Administering Mobile Applications

This document discusses how to administer mobile applications. Topics include:

- [Section 4.1, "Overview"](#)
- [Section 4.2, "Listing Applications"](#)
- [Section 4.3, "Modifying Application Properties"](#)
- [Section 4.4, "Suspending Mobile Applications"](#)
- [Section 4.5, "Resuming Mobile Applications"](#)
- [Section 4.6, "Uploading Applications to the Mobile Server Repository"](#)
- [Section 4.7, "Removing Applications from the Mobile Server"](#)
- [Section 4.8, "Modifying Data Subsetting Parameters"](#)
- [Section 4.9, "Modifying Registry Entries"](#)
- [Section 4.10, "Selecting Application Files for Public Use"](#)
- [Section 4.12, "Adding Bookmarks"](#)
- [Section 4.11, "Adding Web Application Archive \(WAR\) Files"](#)

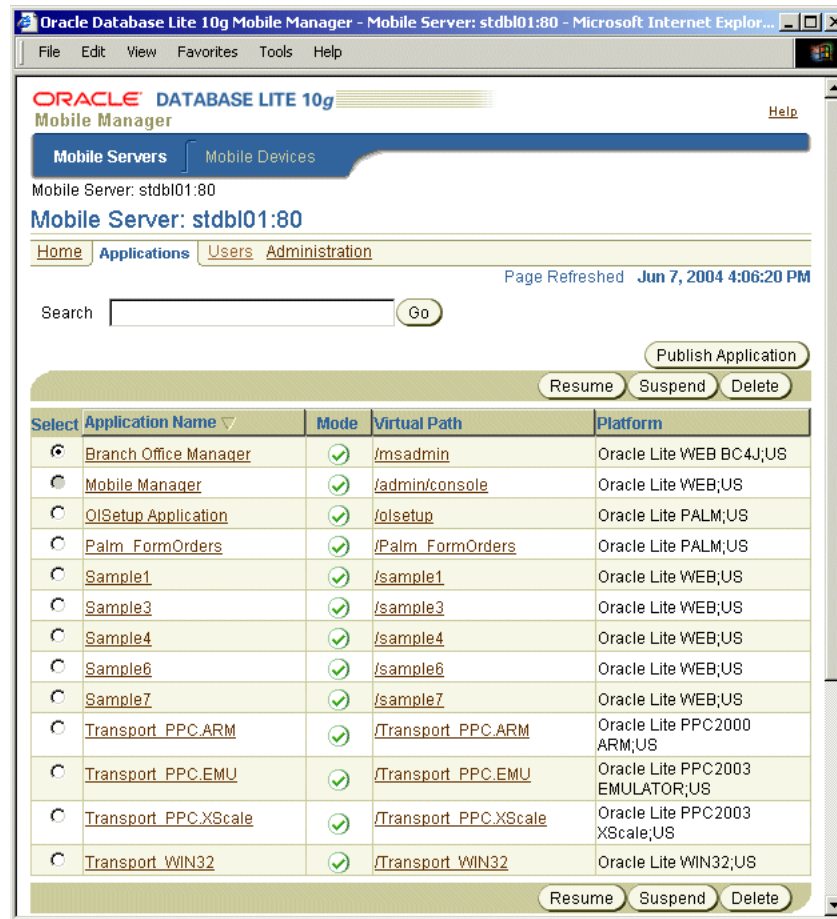
4.1 Overview

Administering mobile applications includes tasks such as listing your web applications, creating database tables, uploading your applications to the mobile server repository, modifying application properties, and so on. The following sections provide a sequential description of tasks that you must perform to administer mobile applications.

4.2 Listing Applications

This section enables you to list mobile applications. To display a list of applications, login to the Mobile Server and click the **Mobile Manager** link in the workspace. The Mobile Server Farms page appears. To display your Mobile Server home page, click your Mobile Server link. Click the **Applications** link. As [Figure 4-1](#) displays, the Applications page lists existing applications and their corresponding virtual path.

Figure 4–1 Applications Page



To search applications, enter your application name in the **Application Name** field and click **Go**. The Applications page displays the search result under the Application Name column.

4.3 Modifying Application Properties

To modify application properties, navigate to the Applications page and click the required Application Name link. As Figure 4–2 displays, the chosen application's page contains application properties and database connectivity details.

Figure 4–2 Application Properties Page

The screenshot shows the Oracle Database Lite 10g Mobile Manager interface. The title bar reads 'Oracle Database Lite 10g Mobile Manager -Application: Branch Office Manager - Microsoft Intern...'. The menu bar includes File, Edit, View, Favorites, Tools, and Help. The main header displays 'ORACLE DATABASE LITE 10g Mobile Manager' with a 'Help' link. Below this, a navigation bar shows 'Mobile Servers' and 'Mobile Devices'. The current page is 'Application: Branch Office Manager'. A sub-navigation bar includes 'Properties', 'Access', 'Data Subsetting', 'Files', and 'Add War File'. A status bar at the top right indicates 'Page Refreshed Jun 7, 2004 4:08:40 PM'.

The 'General' section features a 'Suspend' button, a 'Resume' button, and a status icon (a green checkmark in a blue box). The status is 'Running', the virtual path is '/msadmin', and the published time is 'Jun 3, 2004 12:05:42 PM'.

The 'Application Properties' section contains the following fields:

- Application Name: Branch Office Manager (text input)
- Application Description: Branch Office Manager (text area)
- Publication Name: (text input)
- Platform Name: Oracle Lite WEB BC4J;US (dropdown menu)

The 'Database Connectivity' section contains the following fields:

- Maximum Database Connections: 10 (text input)
- Connection Sharing: NO (dropdown menu)
- Database Username: dummy (text input)
- Database Password: (text input)

At the bottom right, there are three buttons: 'Revert', 'Remove', and 'Apply'.

As described in [Table 4–1](#), enter the modified application properties in the corresponding fields of the Application Properties and Database Connectivity sections.

Table 4–1 Application Properties Page Description

Field	Description
Application Name	name of your mobile application
Application Description	a brief description of your mobile application
Publication Name	publication name of your mobile application
Platform Name	Select the appropriate platform name from the platform name list.
Maximum Database Connections	number of maximum database connections used by your mobile applications
Database User Name	name of the database user
Database Password	password of the database user

To retain the modified application properties, click **Save**.

To remove the application, click **Remove**.

To reset the Application Properties page, click **Reset**.

4.4 Suspending Mobile Applications

To suspend mobile applications, navigate to the Properties page and click **Suspend**. The Mobile Manager seeks your confirmation before suspending the application. Click **Yes**. You are returned to the applications page. As an indicator, the status field displays "Suspended".

4.5 Resuming Mobile Applications

To resume a mobile application that has been suspended, select the application and click **Resume**. The Mobile Manager resumes the chosen application.

4.6 Uploading Applications to the Mobile Server Repository

After building mobile applications and packaging them into a .jar file using the Packaging Wizard, you can upload mobile applications to the Mobile Server Repository.

To upload mobile applications into the Mobile Server Repository on the Oracle database server, navigate to the Applications page and select the mobile application. Click **Publish Application**. As Figure 4–3 displays, the Publish Application page appears.

Figure 4–3 Publish Application Page

The screenshot shows a web browser window titled "Oracle Database Lite 10g Mobile Manager - Publish Application - Microsoft Internet Explorer". The page has a blue header with the Oracle logo and "DATABASE LITE 10g Mobile Manager". Below the header, there are tabs for "Mobile Servers" and "Mobile Devices". The "Mobile Servers" tab is active, showing a breadcrumb "Mobile Server: stdb101:80 > Publish Application". The main content area is titled "Publish Application" and contains the following fields and controls:

- "Repository Directory:" followed by a text input field.
- "My Application file is a:" followed by two radio buttons:
 - ☐ JDeveloper WAR file
 - ☒ Packaging Wizard JAR file
- "Virtual path (WAR file only):" followed by a text input field.
- "Overwrite the existing application:" followed by a checked checkbox.
- "WAR or JAR File:" followed by a text input field and a "Browse..." button.
- At the bottom right, there are "Cancel" and "Upload" buttons.
- At the bottom left, there is a copyright notice: "Copyright © 1997-2004, Oracle. All rights reserved."
- A "Help" link is located in the top right corner.

As described in Table 4–2, enter data in the Publish Application page.

Table 4–2 Publish Application Page Description

Field	Description
Repository Directory	Directory name for your mobile applications. The specified directory name indicates the location of your mobile applications in the Mobile Server Repository. For example, <i>MySample</i> . Note: If you leave the Repository Directory field blank, the Mobile Server automatically creates a directory using the value of the virtual path of your applications.
JDeveloper WAR File	To upload a WAR file, select this option and enter its virtual path in the corresponding field.
Packaging Wizard JAR File	To upload a Packaging Wizard JAR file, select this option.
Overwrite the existing application	To overwrite existing applications with the same virtual path in the Mobile Server Repository, select the Overwrite the existing application check box.
WAR File	To locate the WAR or JAR file, click Browse.

To publish your mobile applications to the Mobile Server Repository, click **Upload**.

4.6.1 Creating Database Tables

As an Administrator, you must ensure that the relevant database tables exist in the Oracle database before publishing mobile applications. Using the SQL command `CREATE TABLE`, you can create tables in the Oracle database. Any table that is used for an updatable snapshot must have the `PRIMARY KEY` constraint.

It is recommended that all records be inserted into your database tables before publishing your mobile application.

4.7 Removing Applications from the Mobile Server

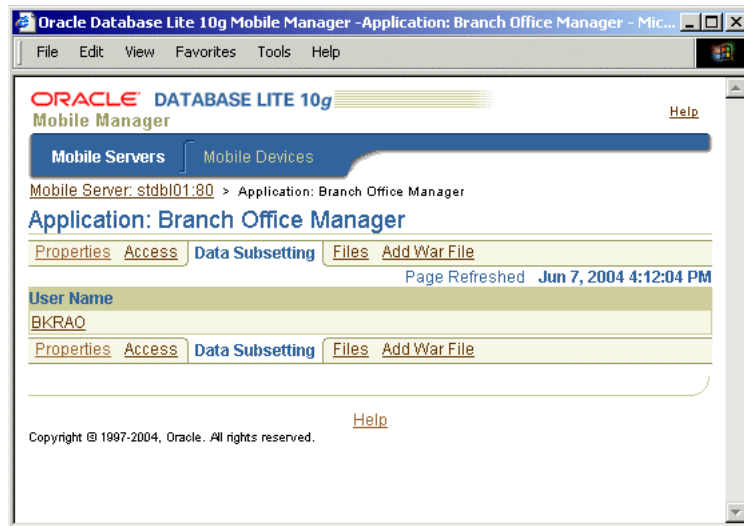
Removing an application from the Mobile Server does not delete all application files and servlets from the Mobile Server system. The application's servlets remain loaded until the last user with access to the Mobile Server applications logs off from the Mobile Server. Once the last user has logged off from the Mobile Server, the administrator can use the `shell` utility in the Mobile Server and delete all application files.

To remove an application from the Mobile Server, navigate to the Applications page and select the application that you want to remove. Click **Delete**. The Mobile Manager seeks your confirmation. Click **Yes**. You are returned to the Applications page.

4.8 Modifying Data Subsetting Parameters

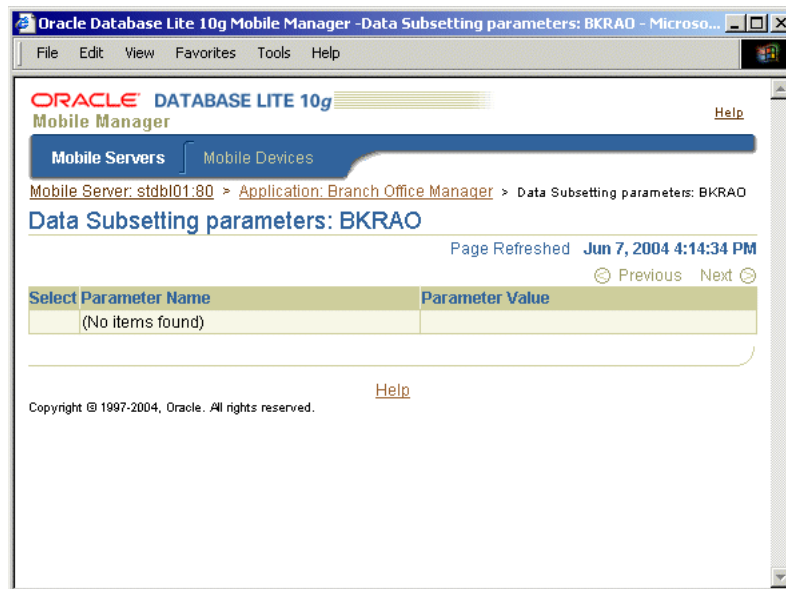
The Mobile Manager enables administrators to view and edit data subsetting parameters in a mobile application. To view and edit data subsetting parameters, navigate to the Applications page and click the application for which you need to view and edit data subsetting parameters. The Applications Properties page appears. Click the **Data Subsetting** link. As [Figure 4–4](#) displays, the Data Subsetting page appears.

Figure 4–4 Data Subsetting Page



The Data Subsetting page lists user names and their associated platforms. Under the User Name column, click the user's name for whom you would like to set data subsetting parameters. As Figure 4–5 displays, the Data Subsetting Parameters page appears and lists the parameter name and value.

Figure 4–5 Data Subsetting Parameters



Enter the changed parameter value and click **Save**. You are returned to the Data Subsetting Parameters page. To reset the Data Subsetting Parameters page, click **Reset**.

4.9 Modifying Registry Entries

To modify registry entries, locate the file `webtogo.ora` and enter your registry value against the `REGISTRY_TAB` parameter.

4.10 Selecting Application Files for Public Use

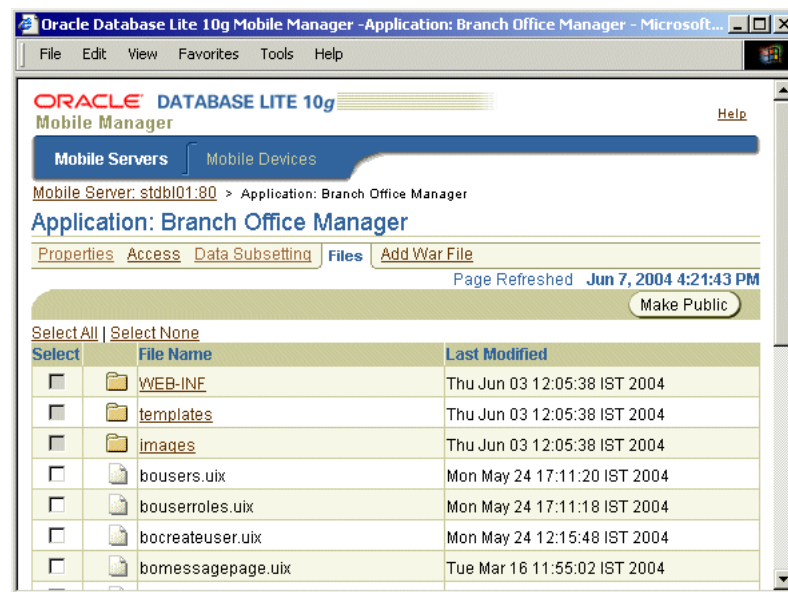
The Mobile Manager enables the Mobile Server administrator to designate specific application files as public files. This feature enables Mobile Clients for Web-to-Go and Branch Office users to download and use these files.

On an application level, application files are available as public files. Every Branch Office or Mobile Client for Web-to-Go to which an application is synchronized will have the same public files available for download. Users can download public files from the Branch Office or Mobile Client for Web-to-Go through the following URL.

`http://<client>/public/download`

To select application files, navigate to the Applications page and click the required application link. The Applications home page appears. Click the **Files** link. As [Figure 4-6](#) displays, the Files page lists application files that are assigned for public use.

Figure 4-6 Files Page



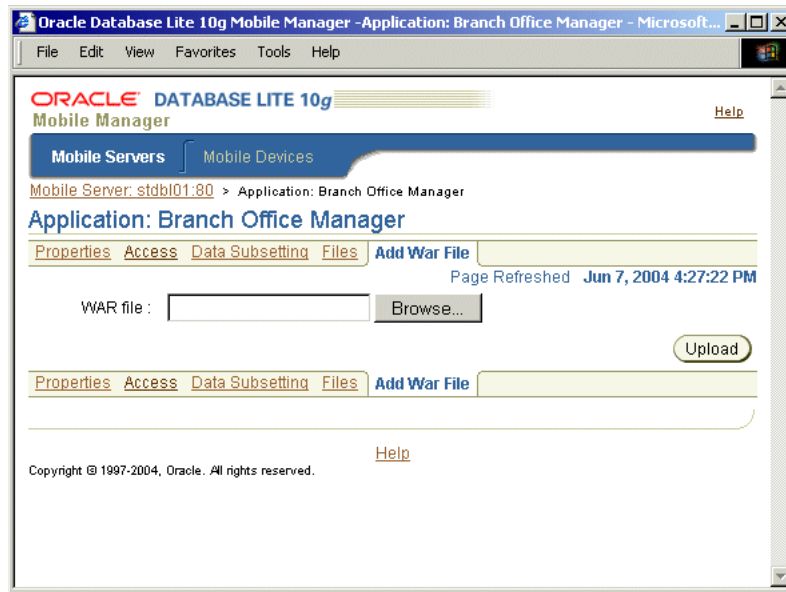
The application file check boxes that enable file selection indicate that the application files have been assigned for public use. Check boxes that are not enabled for selection indicate that the application files are not assigned for public use. Select the check box against the application file that you need for public use and click **Make Public**. You are returned to the Files page.

4.11 Adding Web Application Archive (WAR) Files

Using the Mobile Manager, you can add WAR files to your mobile applications. In accordance with J2EE specifications, you can add Web components to a J2EE application in a package called a Web Application Archive (WAR). The WAR file is basically a JAR file with a few extra rules regarding what data is included within the file. It contains all files that make up a web application including other resources.

To add a WAR file, navigate to the Applications page and click the required application link. The Application Properties page appears. Click the **Add WAR File** link. As [Figure 4-7](#) displays, the Add WAR File page appears.

Figure 4–7 Add WAR File Page

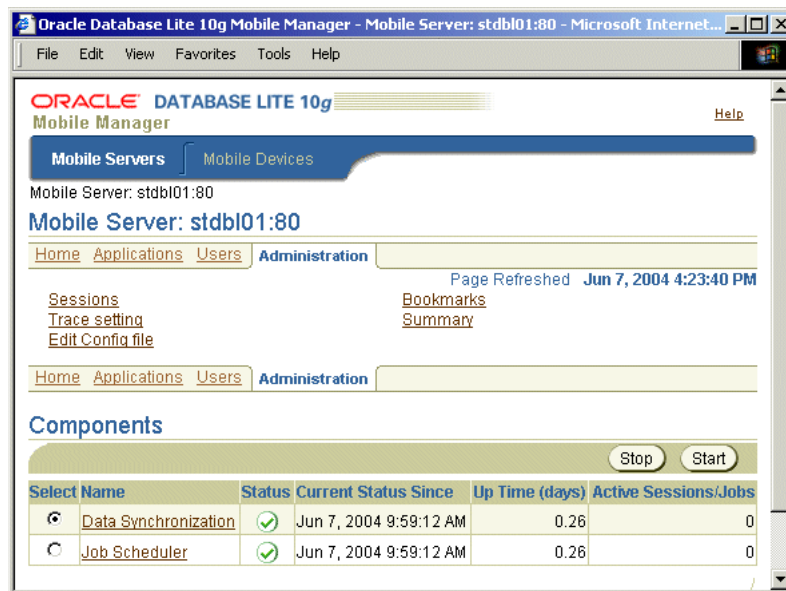


To upload the WAR file, click **Browse** and locate the WAR file. Click **Upload**. You are returned to the Add WAR File page.

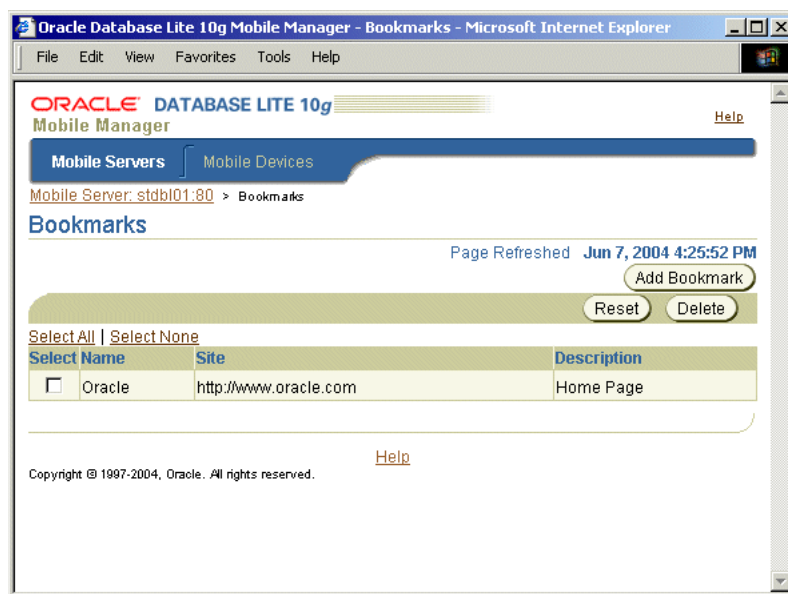
4.12 Adding Bookmarks

To add bookmarks to your mobile applications, click the **Administration** link. As [Figure 4–8](#) displays, the Administration page appears.

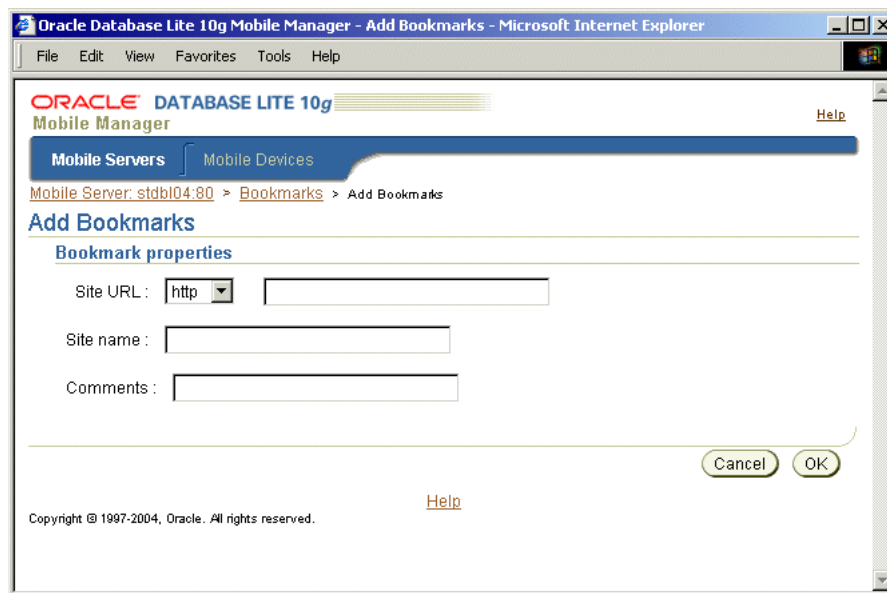
Figure 4–8 Administration Page



Click the **Bookmarks** link. As [Figure 4–9](#) displays, the Bookmarks page appears.

Figure 4–9 Bookmarks Page

Click **Add Bookmark**. As Figure 4–10 displays, the Add Bookmarks page appears.

Figure 4–10 Add Bookmarks Page

Enter data under the Bookmark Properties section as described in Table 4–3 and click **Save**. You are returned to the Mobile Server Bookmarks page which lists your bookmark.

Table 4–3 *Bookmark Properties Description*

Field	Description
Site URL	Web site URL of your mobile application. Choose the appropriate protocol from the list displayed. For example, to indicate a web site address, choose http. To indicate a secure web site address, choose https. To indicate a file transfer site address, choose ftp.
Site Name	Web site name of your mobile application. For example, www.oracle.com.
Comments	Brief description of the web site

Deleting Bookmarks

To delete bookmarks, navigate to the Bookmarks page and select the Bookmark that you want to delete. Click **Delete**.

To reset the bookmarks page, click **Reset**.

Administering Users

This chapter describes how to administer users and mobile applications using the Mobile Manager. Topics include:

- [Section 5.1, "Overview"](#)
- [Section 5.2, "Standalone Mode"](#)

5.1 Overview

The installation option of Oracle Database Lite determines user administration methods that are available to a Mobile Server administrator. As an administrator, you can administer users while running the Mobile Server in Standalone Mode.

5.2 Standalone Mode

In standalone mode, all information relevant to group users and individual users such as user names and passwords, are extracted from the Users table in the Mobile Server Repository. In this mode, the Mobile Server conducts the authentication process.

The following sections discuss how to administer users in Standalone mode. Topics include:

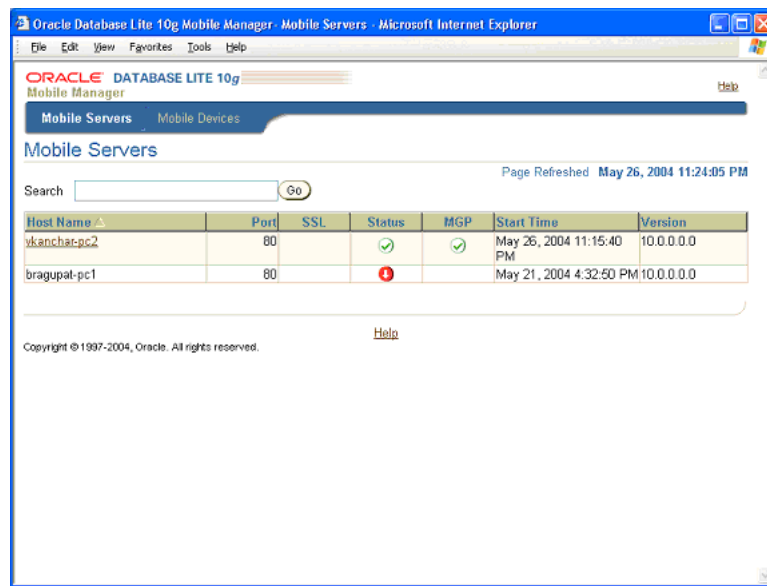
- [Section 5.2.1, "Displaying User Groups and Individual Users"](#)
- [Section 5.2.2, "Adding New Groups"](#)
- [Section 5.2.3, "Adding New Users"](#)

5.2.1 Displaying User Groups and Individual Users

This section enables you to display existing user groups and individual users.

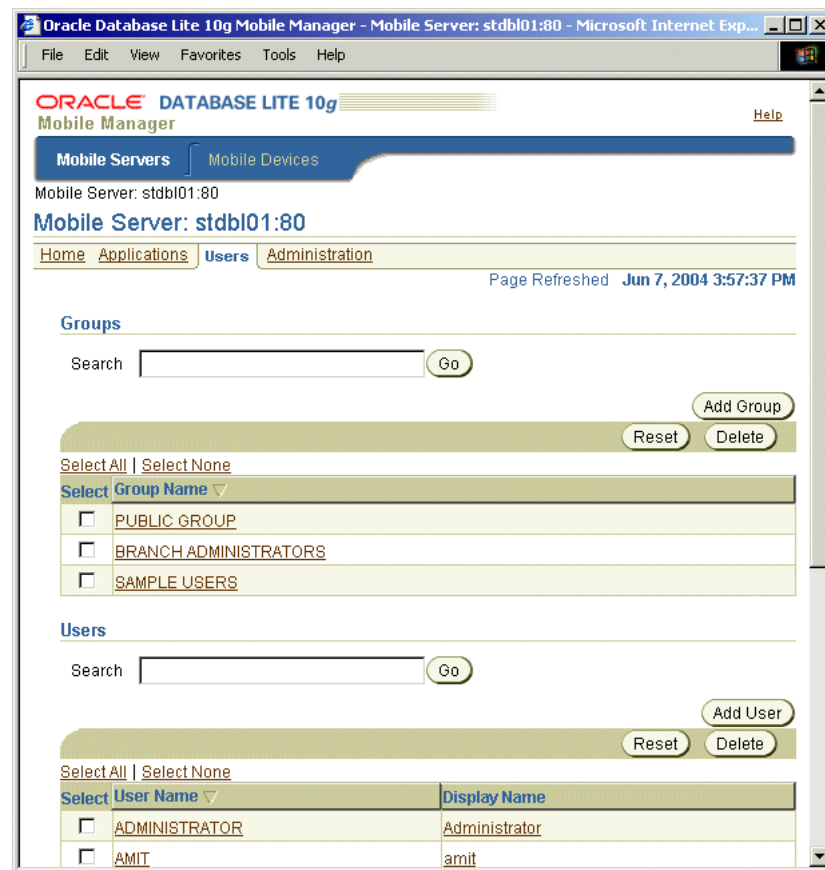
To display individual users, logon to the Mobile Manager and click the **Mobile Manager** link in the Workspace. As displayed in [Figure 5-1](#), the Mobile Server Farms page appears.

Figure 5–1 Mobile Server Farms Page



Click your Mobile Server name link. Your Mobile Server home page appears. Click the **Users** link. As Figure 5–2 displays, the Users page lists existing groups and individual users.

Figure 5–2 Users Page



Searching Group Names or User Names

To search for a group name or individual user name, enter the group name or user name in the **Search** field and click **Go**. The Users page displays the search result under the Group Name or User Name column.

5.2.2 Adding New Groups

As an administrator, you can add a new group that accesses the Mobile Server in Standalone mode. The Users table in the Mobile Server Repository contains information about new users.

To add a new group, navigate to the Users page and click **Add Group**. As Figure 5–3 displays, the Add Group page appears and lists the requisite criteria to register user group properties.

Figure 5–3 Add Group Page

The screenshot shows a web browser window titled "Oracle Database Lite 10g Mobile Manager - Add Group - Microsoft Internet Explorer". The page has a blue header with the Oracle logo and "DATABASE LITE 10g Mobile Manager". Below the header, there are tabs for "Mobile Servers" and "Mobile Devices". The "Mobile Servers" tab is selected, and the breadcrumb "Mobile Server: stdbl01:80 > Add Group" is visible. The main heading is "Add Group". There is a text input field labeled "Group Name :". At the bottom right, there are "Cancel" and "OK" buttons. A "Help" link is located at the bottom center. The footer text reads "Copyright © 1997-2004, Oracle. All rights reserved."

Enter the new group name in the **Group Name** field and click **OK**.

5.2.3 Adding New Users

To add a new user, navigate to the Users page and click **Add User**. As [Figure 5–4](#) displays, the Add User page appears and lists the requisite criteria to register user properties.

Figure 5–4 Add User Page

The screenshot shows a web browser window titled "Oracle Database Lite 10g Mobile Manager - Add User - Microsoft Internet Explorer". The page has a blue header with the Oracle logo and "DATABASE LITE 10g Mobile Manager". Below the header, there are tabs for "Mobile Servers" and "Mobile Devices". The "Mobile Servers" tab is selected, and the breadcrumb "Mobile Server: stdbl01:80 > Add User" is visible. The main heading is "Add User". There are four text input fields labeled "Display Name :", "User Name :", "Password :", and "Password Confirm :". Below these is a dropdown menu labeled "Privilege :" with "USER" selected. At the bottom right, there are "Cancel" and "OK" buttons. A "Help" link is located at the bottom center. The footer text reads "Copyright © 1997-2004, Oracle. All rights reserved."

To register user properties for new users, [Table 5–1](#) describes values that must be entered in the Add User page.

Table 5–1 Add User Page Description

Field	Description
Display Name	Name used to display as Mobile Server user name.
User Name	Name used to logon to the Mobile Server.
Password	Password used to logon to the Mobile Server.
Password Confirm	To confirm the above mentioned password, re-enter your password.
Privilege	<p>Lists available privileges for the Mobile Server user.</p> <ul style="list-style-type: none">■ The Administrator option provides privileges to modify Mobile Server resources.■ The Organizer option enables users to publish applications.■ The User option provides access for registered users to the Mobile Server.

Enter the user information as described in the above table and click **OK**.

Note: User names and passwords can only contain single-byte characters and cannot contain characters such as ', ", @ or %.

Deleting Groups or Individual Users

As an administrator, you can delete groups or individual users from the system. To permanently delete groups or individual users from the system, select the **Delete** check box against the group name or individual user name that you want to delete, and click **Delete**. The Mobile Manager seeks your confirmation to delete the chosen group or user name. Click **Yes**. You will be returned to the Users page.

To reset the group names and individual user names, click **Reset**.

Provisioning

This document enables the Mobile Server Administrator to grant access privileges to mobile application developers and manage snapshots. Topics include:

- [Section 6.1, "Managing Access Privileges"](#)
- [Section 6.2, "Managing Snapshots"](#)

6.1 Managing Access Privileges

The following sections describe the access feature of the Mobile Server. Topics include:

- [Section 6.1.1, "Granting or Revoking Application Access to Users and Groups"](#)
- [Section 6.1.2, "Including or Excluding Users from Group Based Access"](#)

6.1.1 Granting or Revoking Application Access to Users and Groups

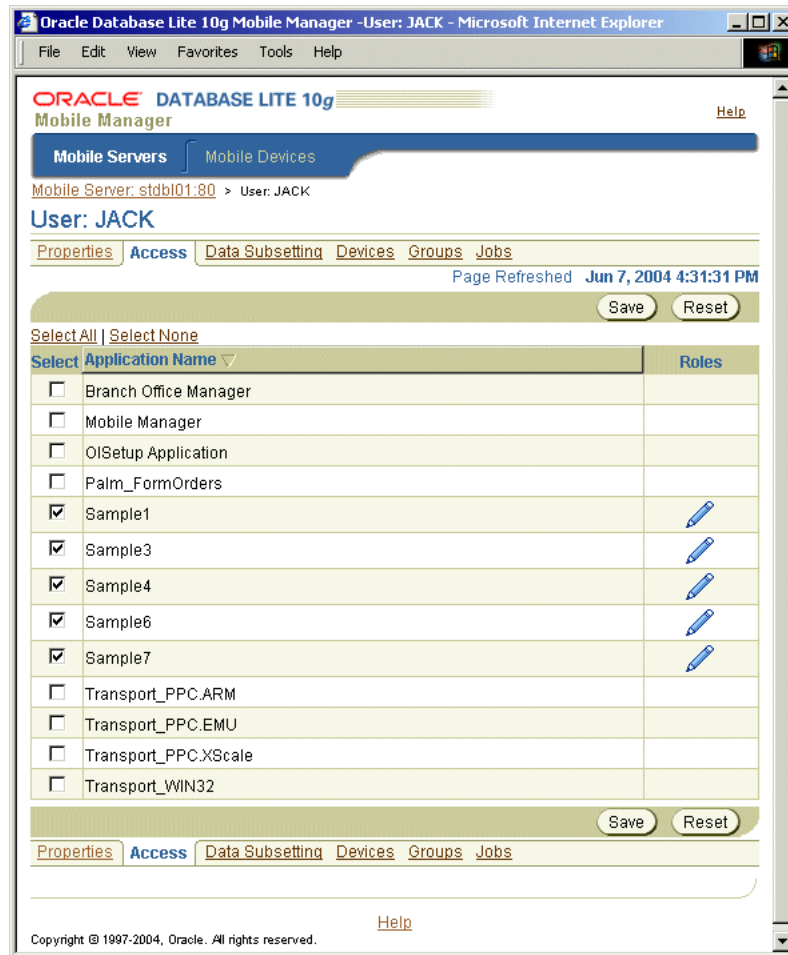
This section enables the Administrator to grant or revoke application access to users and groups. Topics include:

- [Granting Application Access to Users](#)
- [Revoking Application Access to Users](#)

Granting Application Access to Users

To grant application access to users, login to the Mobile Server. Using the Mobile Manager, navigate to the Users page and click the User Name link. The User Properties page appears. Click the Access link. As [Figure 6-1](#) displays, the Access page displays a list of available applications. Select the applications that you want to grant access to and click Save.

Figure 6–1 Granting Application Access



Revoking Application Access to Users

To revoke application access to users, clear the check box displayed against an application name and click Save.

Note: Granting application access to an entire group gives each user in the group, access to the application. However, you can exclude certain users from accessing the application without removing them from the group. For more information, see the following section.

6.1.2 Including or Excluding Users from Group Based Access

This section enables the Administrator to include or exclude users from group based access. Topics include:

- [Granting Group Based Access to Users](#)
- [Revoking Group Based Access to Users](#)

Using the Mobile Manager, you can modify group based access privileges to include or exclude users requiring access to mobile applications. To modify group based access

privileges, click the **Users** link. The Users page lists existing groups and individual users.

Granting Group Based Access to Users

To grant group based access to users, navigate to the Users page and click the **User Name** link. The corresponding User Properties page appears. Click the **Groups** link. Select the group name that you want to include the user into and click **Save**.

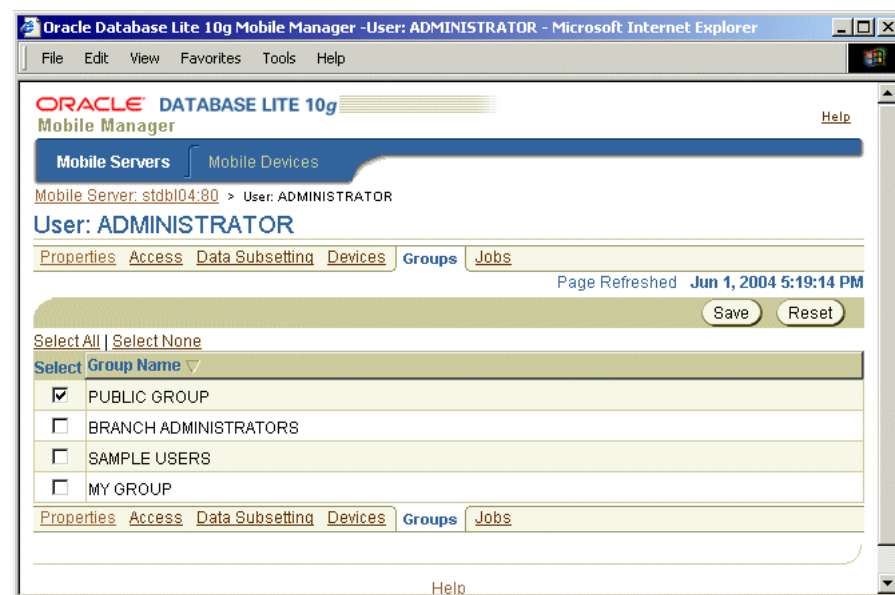
Note: Existing users with privileges for group based access only can be excluded from group based access.

Revoking Group Based Access to Users

To revoke group based access to users, navigate to the Users page and click the **User Name** link. The corresponding User Properties page appears. Click the **Groups** link. Clear the group name that you want to remove the user from and click **Save**.

Figure 6–2 displays the Add Group page.

Figure 6–2 Add Group Page



6.2 Managing Snapshots

A snapshot is a full set or a subset of rows of a table or view at a pre-determined time. It is created by executing an SQL query against the base table. Snapshots are either read-only or updatable. They vary in complexity.

This section describes how to manage snapshots. Topics include:

- [Section 6.2.1, "Read-only Snapshots"](#)
- [Section 6.2.2, "Updatable Snapshots"](#)
- [Section 6.2.3, "Refreshing a Snapshot"](#)
- [Section 6.2.4, "Snapshot Template Variables"](#)

6.2.1 Read-only Snapshots

Read-only snapshots are used for querying purposes. Changes made to the master table are replicated to the snapshot by the Mobile Client.

6.2.2 Updatable Snapshots

Updatable snapshots provide updatable copies of a master table. You can define updatable snapshots to contain a full copy of a master table or a subset of rows in the master table that satisfy a value-based selection criteria. You can make changes to the snapshot which the Mobile Sync propagates back to the master table.

A snapshot can only be updated when all the base tables that the snapshot is based on have a primary key. If the base tables do not have a primary key, a snapshot cannot be updated and becomes read-only.

6.2.3 Refreshing a Snapshot

Your snapshot definition determines whether an updatable snapshot uses the complete or fast refresh method. The complete refresh method recreates the snapshot every time it is refreshed. The fast refresh method refreshes the snapshot's existing data. In general, the simpler your snapshot definition, the faster it is updated. For more information on the fast refresh method, see the *Oracle Database Lite Developer's Guide*.

6.2.4 Snapshot Template Variables

Snapshots are application-based. Every client that goes offline uses the same snapshot definition. As a result, every client downloads the same application data. In some cases, you may want to specify the data that your application downloads for each user. You can accomplish this by using snapshot templates.

A snapshot template is an SQL query that contains data subsetting parameters. A data subsetting parameter is a colon (:) , which is followed by an identifier name. For example:

```
:var1
```

When the Mobile Client creates snapshots on the client machine, it replaces the snapshot variables with user-specific values. By specifying different values for different users, you can control the number of rows returned by the query.

You can use the Packaging Wizard to specify a snapshot template variable in the same way that you create a snapshot definition for any platform.

Data subsetting parameters cannot be part of a string and therefore should not be enclosed in single quotation marks ('). If you want to specify a string as the value of the data subsetting parameter, the string itself must contain single quotation marks. You can specify the values for the template variables using the Mobile Manager.

Examples

The following examples specify a different value for every user. By specifying a different value for every user, the Administrator can control the behavior and output of the snapshot template.

Example 1

Snapshot Template: `select * from emp where deptno = :dno`

[Table 6–1](#) provides a sample set of snapshot query values specified for separate users.

Table 6–1 Snapshot Query Values for Separate Users

User	Value	Snapshot Query
John	10	select * from emp where deptno = 10
Jane	20	select * from emp where deptno = 20

Example 2

Snapshot Template: select * from emp where ename = :ename

[Table 6–2](#) provides another sample snapshot query value.

Table 6–2 Snapshot Query Value for User Names

User	Value	Snapshot Query
John	'KING'	select * from emp where ename = 'KING'

For more information on specifying data subsetting values using the Mobile Manager, see [Chapter 4, "Administering Mobile Applications", Section 4.8, "Modifying Data Subsetting Parameters"](#).

Administering the Mobile Server

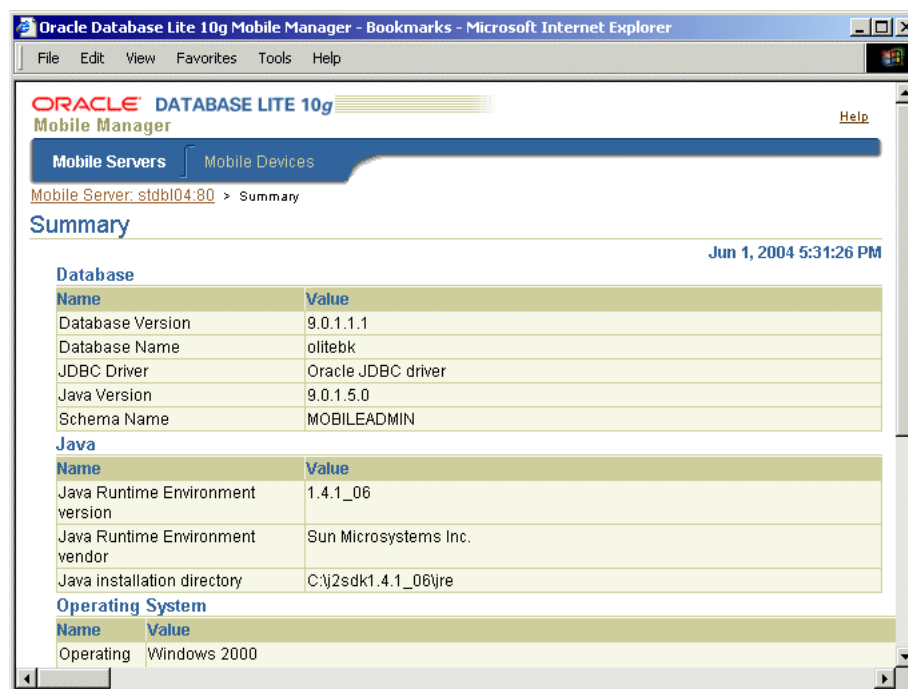
This chapter enables users to administer the Mobile Server. Topics include:

- [Section 7.1, "Viewing System Status Reports for the Server"](#)
- [Section 7.2, "Suspending Applications"](#)
- [Section 7.3, "Resuming Applications"](#)
- [Section 7.4, "Viewing Active User Sessions"](#)

7.1 Viewing System Status Reports for the Server

The Mobile Manager enables users to view system status reports for the Mobile Server. To view system status reports, click the **Administration** link and click the **Summary** link. As [Figure 7-1](#) displays, the Summary page lists Database, JRE, and Operating System details.

Figure 7-1 Summary Page



7.2 Suspending Applications

The Mobile Manager enables users to suspend the Mobile Server for maintenance purposes. At this stage, the Mobile Server terminates all currently active sessions on the server and restricts access only to users with administrative privileges.

To suspend the Mobile Server, navigate to the applications page and click the **Suspend** button. The Mobile Server seeks your confirmation. Click **Yes**.

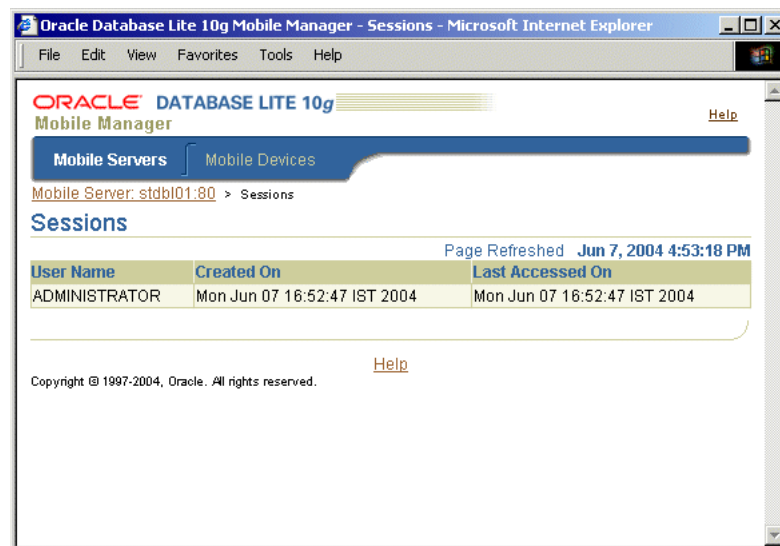
7.3 Resuming Applications

To resume the Mobile Server, navigate to the Application Properties page and click the **Resume** button.

7.4 Viewing Active User Sessions

The Mobile Manager enables administrators to display a list of all users that are connected to the Mobile Server at any given time. To view a report on active user sessions, navigate to the Administration page and click **Sessions**. As [Figure 7-2](#) displays, the Sessions page lists user names, date and time of creating the user's session, and the date and time of the last session.

Figure 7-2 Sessions Page



Managing Synchronization

The Mobile Server administrator uses the Data Synchronization Manager to manage synchronization tasks. This chapter includes:

- [Section 8.1, "Overview"](#)
- [Section 8.2, "The Data Synchronization Manager: An Introduction"](#)
- [Section 8.3, "Managing the Sync Service"](#)
- [Section 8.4, "Monitoring and Analyzing Sync Performance"](#)
- [Section 8.5, "Administering the Synchronization Service"](#)
- [Section 8.6, "Browsing the Sync Repository"](#)
- [Section 8.7, "Scheduling MGP Cycles to Run Inside the Mobile Server"](#)
- [Section 8.8, "Running the Message Generator and Processor \(MGP\) from the Command Line"](#)
- [Section 8.9, "Monitoring Synchronization Using SQL Scripts"](#)

8.1 Overview

The Mobile Server uses synchronization to replicate data between Oracle Database Lite databases (including those for Web-to-Go, Win32, Palm, and Windows CE platforms) and the Oracle database. For example, in a Web-to-Go application, when the user switches to offline mode, Web-to-Go replicates the user's applications and data from the Oracle database to Oracle Database Lite. When the user switches back to online mode, Web-to-Go replicates any Oracle Database Lite changes to the Oracle database.

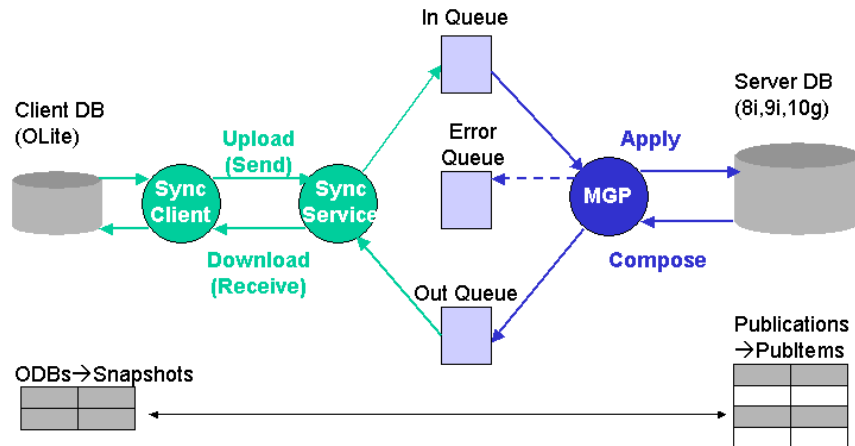
8.2 The Data Synchronization Manager: An Introduction

The Oracle 10g Data Synchronization Manager provides a graphical user interface to manage the Sync service, monitor and analyze Sync service performance, administer Sync service configuration and tracing, browse the Sync service repository for publication and subscription information, and monitor and analyze MGP performance. These functionalities correspond to five tabs on the Data Synchronization home page named Home, Performance, Administration, Repository, and MGP.

The Home tab stores Sync service details and the Performance tab stores sync performance details. The MGP tab stores information related to MGP functions and performance. The Repository tab stores client information related to the In-Queues and Out-Queues. To set the configuration parameters for the whole system, the Administration tab provides the necessary controls.

As displayed in [Figure 8–1](#), to view an overview of the Data Synchronization architecture, login to the Mobile Server and navigate to the Data Synchronization home page. Click **Architecture**.

Figure 8–1 Data Synchronization Architecture



The Oracle Lite database contains a subset of data that is stored in the main Oracle database, which is the back-end database. This subset is stored as a snapshot in the Oracle Lite database. Unlike a base table, a snapshot tracks changes made to it in a change log. As a user, you can make changes in the Oracle Lite database while the device is disconnected, and can then connect to the back-end Oracle database for synchronizing refreshed data within a snapshot.

The Data Synchronization component in the Mobile Server receives or uploads client transactions from the Oracle Lite database and applies these transactions to the back-end Oracle database. The Data Synchronization component composes the server-side changes for a client into transactions and sends or downloads them to the client Oracle Lite database.

The Data Synchronization component employs an asynchronous model and assigns the synchronization task to two sub-components called Sync service and MGP.

8.3 Managing the Sync Service

The Synchronization service (Sync service) feature is an HTTP servlet that listens to client synchronization requests. During every synchronization session, the Sync service receives or uploads client transactions in the Oracle Lite database and places them within the in-queues. The Sync service then sends or downloads server-side transactions from the out-queues to the client Oracle Lite database.

The Synchronization Manager enables users to view a table of active synchronization sessions, in which synchronization clients are connected with the Sync service and are engaged in uploading or downloading transactions. After the client disconnects, synchronization session details are stored in the synchronization history page if the SYNC_HISTORY instance parameter is set to TRUE, which is the default parameter value.

The Data Synchronization home page enables administrators to manage synchronization (Sync) service tasks such as starting the Sync service and checking for alerts that are registered in the Sync service. Using this page, administrators can

manage active synchronization sessions and the synchronization history list. Topics include:

- [Section 8.3.1, "Starting/Stopping the Sync Service"](#)
- [Section 8.3.2, "Checking Sync Service Alerts"](#)
- [Section 8.3.3, "Managing Active Sync Sessions"](#)
- [Section 8.3.4, "Managing the Session History List"](#)
- [Section 8.3.5, "Displaying Operating System \(OS\) and Java Virtual Machine \(JVM\) Information"](#)

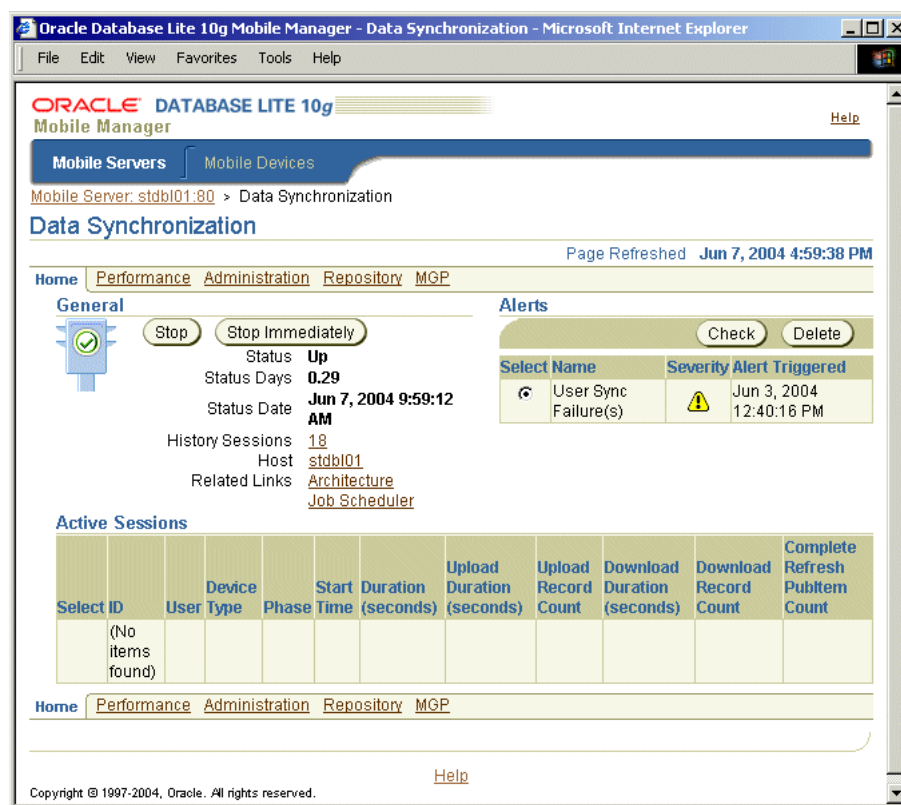
8.3.1 Starting/Stopping the Sync Service

To start the Sync service, navigate to the Data Synchronization home page as follow.

1. Login to the Mobile Server using the appropriate user name and password.
2. Locate the Mobile Server components table, and click **Data Synchronization**. The Data Synchronization home page appears.

The Sync service's default status is Up. As displayed in [Figure 8–2](#), the Data Synchronization home page displays the default status.

Figure 8–2 Data Synchronization Home Page



To stop the Sync service, click **Stop**. The Mobile Server displays a warning message that seeks your confirmation to stop the Sync service. Click **Yes**. You are returned to the Synchronization Service home page.

To stop the Sync service immediately, click **Stop Immediately**.

8.3.2 Checking Sync Service Alerts

The Sync service serves multiple clients that require synchronization. Clients cannot synchronize if the sync service encounters an exception. This occurrence is registered as a critical alert. In this situation, the Sync service displays details of the exception. As the Database Administrator (DBA), you must check such exceptions.

For critical alerts that can be resolved, such as when the database is down, the DBA analyzes and resolves the problem, and then re-starts the synchronization service. Warning alerts are registered if an individual synchronization session fails. In this case, the DBA must check the Sync session details in the Sync history, where the details of the failure are recorded, and determine the reasons for failure. Alternatively, the DBA can report problems that are un-resolvable to Oracle support.

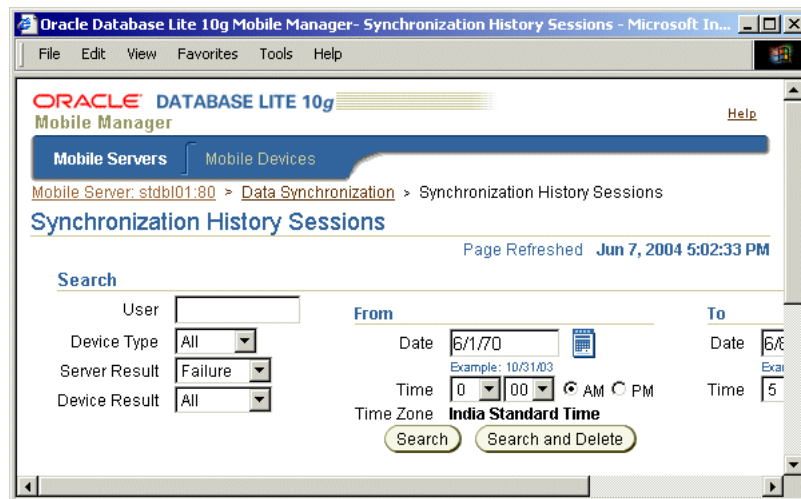
The Data Synchronization home page enables the DBA to check alerts that are registered in the Sync engine. It provides information such as the alert name, degree of severity, time when the alert was triggered, and time when the alert was last checked by a DBA.

The Data Synchronization home page displays Sync service exception alerts. To check other alerts, scroll down to the Alerts table and select the alert that you need to view in the select column. Click **Check**.

As described in [Figure 8–3](#), the Synchronization History Sessions page displays failed and unchecked history sync sessions of User Sync Failure alerts.

The MGP tab displays MGP job exceptions. The MGP Apply/Compose Cycles page displays failed and unchecked History Cycles of MGP User Apply/Compose failures.

Figure 8–3 Synchronization History Sessions Page



To delete an alert, locate the Alerts table on the Data Synchronization home page. Select the alert and click **Delete**. As listed in [Table 8–1](#), sample alert names and types are based on the corresponding severity.

Table 8–1 Alert Types

Name	Type	Severity
Sync Service Exception	Sync Service	CRITICAL
User Sync Failure(s)	Sync Service	WARNING

Table 8–1 (Cont.) Alert Types

Name	Type	Severity
MGP Job Exception	MGP	CRITICAL
MGP User Apply/Compose Failure(s)	MGP	WARNING

8.3.3 Managing Active Sync Sessions

The Active Sessions table on the Data Synchronization home page displays the following active sync session information.

- User Name
- Device Type
- Phase
- Start Time
- Duration
- Upload and Download Duration (in seconds)
- Upload and Download Record Count
- Complete Refresh Publication Item Count

To terminate an active session, perform the following steps.

1. Select the active session that you wish to terminate and click **Kill**. The Mobile Server displays a warning message seeking your confirmation to terminate the active session.
2. Click **Yes**. The Mobile Server displays a confirmation message stating that the chosen session is terminated.
3. Click **OK**. You are returned to the Data Synchronization home page.

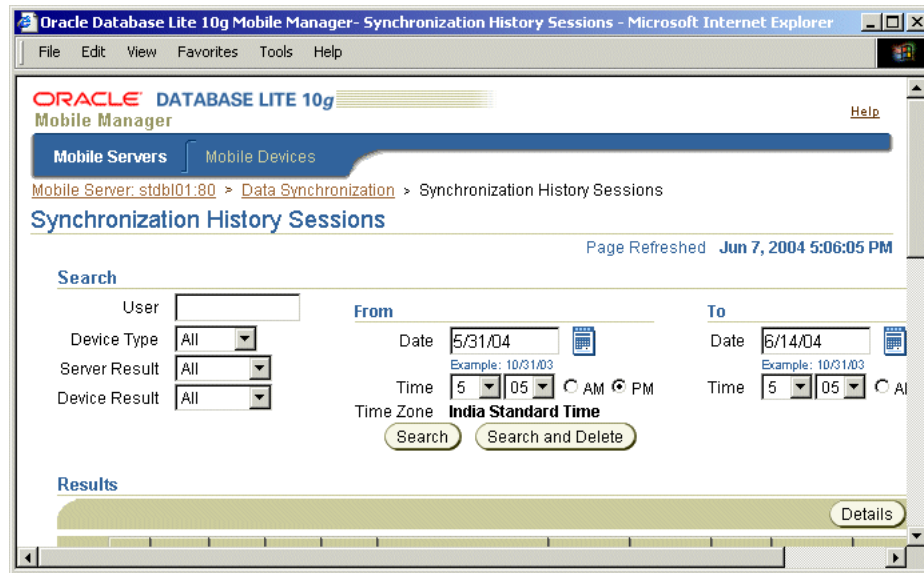
The Active Sessions table on the Data Synchronization home page also displays session details. Select the active session that you wish to view and click **Details**. The Active Sync Session page displays the chosen session's publication items that have been uploaded or downloaded, waiting publication items, records and timing information, and the session trace file.

8.3.4 Managing the Session History List

The Data Synchronization home page displays the total number of sessions that are registered in the Session history list. The administrator can search, sort, and manage the session history list that is based on Session properties.

To display the session history list, click the number hyperlink that is displayed against the Session History. For example, click the number that is displayed as a hyperlink against Session History. As displayed in [Figure 8–4](#), the Synchronization History Sessions page appears.

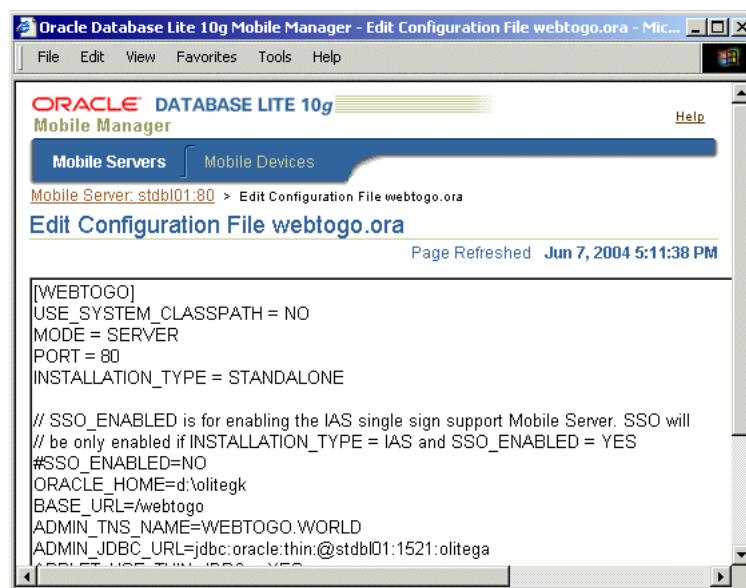
Figure 8–4 Synchronization History Sessions Page



- To search the Session history list, enter your search criteria such as user name, date, and time in the corresponding fields. Click **Search**. The Session History page displays matched sync sessions in the **Results** section.
- To sort matched sync sessions, click the **Header Title** of the sync session item that you wish to sort. For example, to sort sync sessions by user, click the **User** header title.
- To delete a session, select the session that you want to delete and click **Delete**.
- To view the details of a session, select the session and click **Details**. The Sync History Session page displays session details such as publication items that are uploaded or downloaded, records and timing information, and the session trace file.
- The **View** and **Download** links are automatically enabled for viewing or downloading trace files that are available for the chosen session.

8.3.5 Displaying Operating System (OS) and Java Virtual Machine (JVM) Information

To display the OS and JVM info, click the **Host** hyperlink that is displayed against the Host. As displayed in [Figure 8–5](#), the Host page displays host information such as host name, IP address, OS type, and OS user name. The JVM section displays the java classpath, java version, and heap memory size.

Figure 8–5 Host Page

8.4 Monitoring and Analyzing Sync Performance

This section describes how to monitor and analyze sync performance. Topics include:

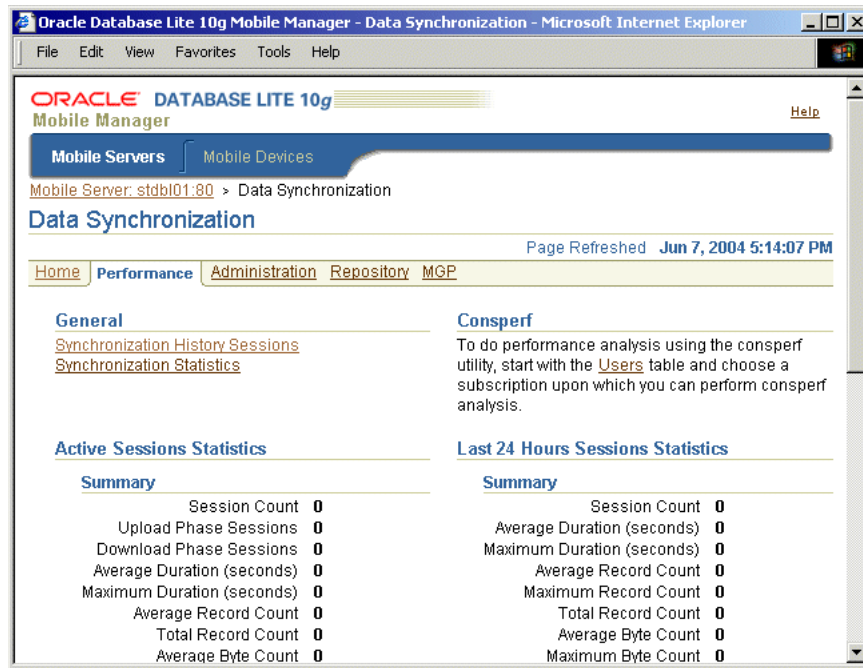
- [Section 8.4.1, "Viewing Sync Statistics"](#)
- [Section 8.4.2, "Analyzing Performance of Publications"](#)

8.4.1 Viewing Sync Statistics

The Performance tab displays statistics of the current session and statistics of history sessions that have occurred in the last 24 hours.

To view Sync Statistics, click the **Performance** tab. As displayed in [Figure 8–6](#), the Performance tab appears.

Figure 8–6 Performance Page



To view additional statistics, click the **Synchronization Statistics** link in the General section of this page.

The Sync Statistics page contains search criteria such as user name, device type, and duration. Specify your criteria in the Search section and click **Go**. The Sync Statistics page displays results such as summary, upload phase, and download phase details.

8.4.2 Analyzing Performance of Publications

The Performance tab enables users to conduct a performance analysis using the Consperf utility.

The Consperf utility profiles Sync service publications. Application developers and administrators use this utility to analyze the performance of publications and identify potential bottlenecks during publication. This tool enables users to perform four primary functions:

1. Generate timing statistics for publications
2. Generate explain plans for publications
3. Automatically tune publication properties
4. Analyze Mobile Server objects for the Cost Based Optimizer

The Performance tab provides enhanced performance analysis and tuning capabilities and is more convenient than its command line counterpart. It enables users to start with a list of clients and choose the required subscription for performance analysis. Users can change parameter values before analyzing performance. The analysis results, which are timing and plan information are stored on the server and can be accessed by viewing the corresponding subscription.

To analyze Consperf, perform the following steps.

1. Navigate to the Performance tab and click **Users** under the Conserpf section. As displayed in Figure 8–7, the Users page displays a list of users.

Figure 8–7 Users Page

Oracle Database Lite 10g Mobile Manager - Users - Microsoft Internet Explorer

File Edit View Favorites Tools Help

ORACLE DATABASE LITE 10g Mobile Manager

Mobile Servers Mobile Devices

Mobile Server: stdb01:80 > Data Synchronization > Users

Users

Page Refreshed Jun 7, 2004 5:15:43 PM

Search [User] Go

Subscriptions

Select	User	In Queue Has Data	Out Queue Has Data	Last Synchronization Time
<input checked="" type="radio"/>	S11U1	No	Yes	Jun 3, 2004 12:00:00 AM
<input type="radio"/>	R	No	Yes	
<input type="radio"/>	JOHN	No	No	Jun 7, 2004 12:00:00 AM
<input type="radio"/>	JANE	No	Yes	
<input type="radio"/>	JACK	No	Yes	Jun 3, 2004 12:00:00 AM
<input type="radio"/>	GAN1086607577	No	Yes	
<input type="radio"/>	GAN1086607505	No	Yes	
<input type="radio"/>	GAN1086607397	No	Yes	
<input type="radio"/>	GAN1086607262	No	No	
<input type="radio"/>	GAN1086607096	No	Yes	

2. Select a user and click **Subscriptions**. The Subscriptions page displays a list of subscriptions for the chosen user.
3. Select a subscription and click **Conserpf Performance Analysis**. The Conserpf Performance Analysis page appears.
4. Click the hyperlink **Set conserpf parameters and launch the conserpf thread**. The Run Conserpf page appears.

The Run Conserpf page associates all the available parameters, their corresponding default values and descriptions. As a user, you can make the necessary changes to the parameter values and click **Run Conserpf**. At this stage, the Conserpf thread is started and the user is returned to the Conserpf Performance Analysis page, which displays information related to the status of Conserpf. Upon completion of the performance analysis, the Conserpf Performance Analysis page displays hyperlinks to results of the analysis.

5. To view performance analysis results conducted by Conserpf, click the hyperlinks **View Timing File** or **View Execution Plan File**.

8.5 Administering the Synchronization Service

The Administration tab enables users to view and change synchronization parameters, trace settings, and trace files. This section describes the following topics:

- [Section 8.5.1, "Managing Configuration Parameters"](#)
- [Section 8.5.2, "Managing Trace Settings and Trace Files"](#)

8.5.1 Managing Configuration Parameters

There are two types of configuration parameters for the Sync service. They are: Shared and Instance. Shared parameters affect all instances. Instance parameters only affect a single instance. The Mobile Server can contain multiple instances that connect to the same repository.

The Sync manager provides a form to view and edit these parameters. It provides user access to current values, default values, and descriptions of these parameters. Users can also view and edit the parameter file directly on the web.

The following sections describe how to view parameter descriptions and edit existing parameter values.

- [Viewing Parameter Descriptions](#)
- [Editing Parameter Values](#)

Viewing Parameter Descriptions

To view parameter descriptions and other additional information, navigate to the Administration tab and click the **Shared Parameters** or **Instance Parameters** hyperlink. Based on your option, the Shared Parameters page or Instance Parameters page appears.

For example, click **Shared Parameters**. The Shared Parameters page displays existing parameters. To view the parameter description and additional information, click **Show**.

Editing Parameter Values

To edit parameter values, click the **Shared Parameters** or **Instance Parameters** hyperlink. Based on your option, the Shared Parameters page or Instance Parameters page appears. In the **New Value** field, edit the value and click **Apply**. Some parameter values do not take effect until the synchronization service is restarted.

Note: The Instance Parameter table only contains the Need Restart column.

8.5.2 Managing Trace Settings and Trace Files

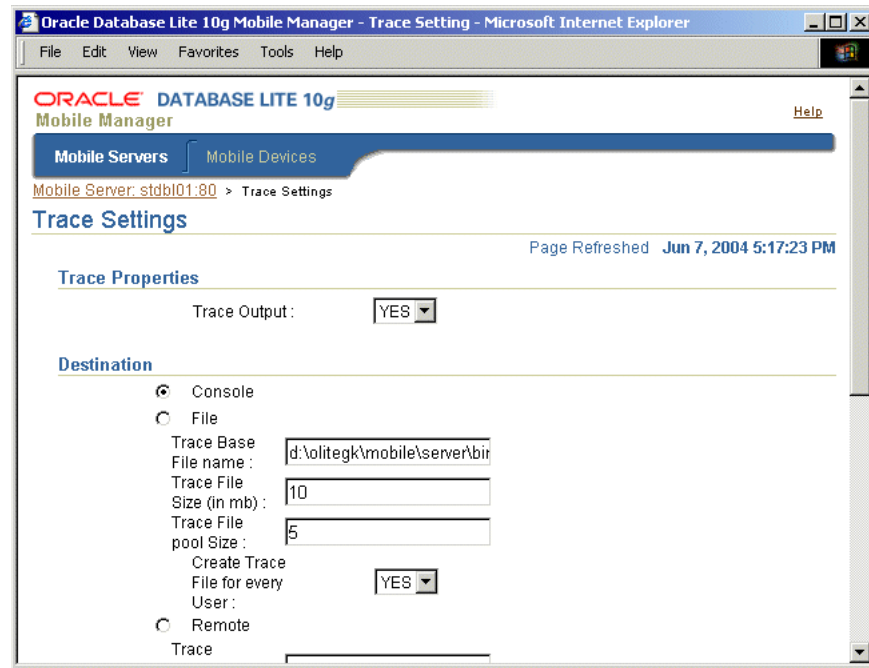
This section describes how to manage trace settings and trace files. Topics include:

- [Managing Trace Settings](#)
- [Managing Trace Files](#)
- [Consolidator Components for Tracing](#)
- [Tracing Levels and Types](#)

Managing Trace Settings

1. To manage trace settings, navigate to the Administration tab and click **Trace Settings**.

As displayed in [Figure 8–8](#), the Trace Settings page appears.

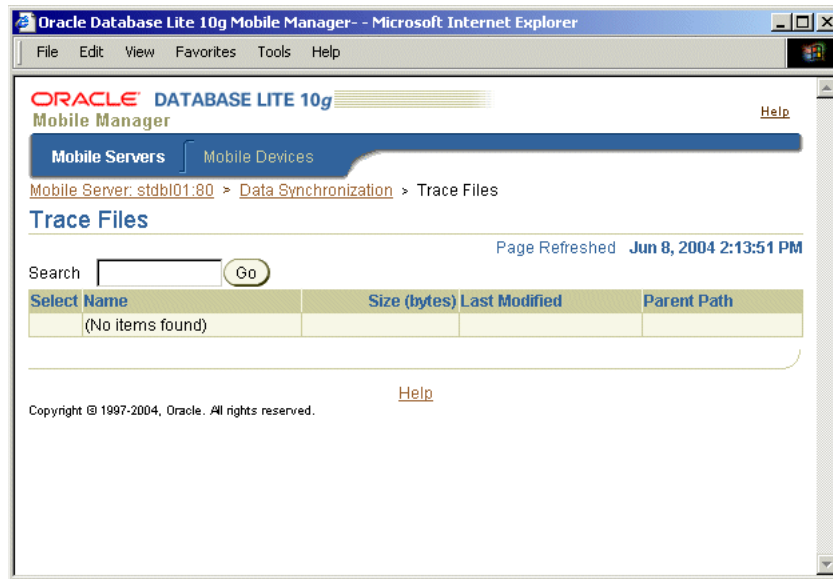
Figure 8–8 Trace Settings Page

2. The Trace Settings page displays five components. To change trace settings, enter the modified data in the corresponding fields and click **Apply**.
3. In the Filter section, select the required **Level** and **Type**. To specify a trace filter for users, enter comma separated user names in the **Users** field. In the Destination section, select **Local Console** to receive the trace file on your local console. To retrieve trace information to a file, select the **File** option. The File Size and File Pool Size fields display the values 10 and 2 by default. If required, enter the modified file size and file pool size. To implement the modified values, click **Apply**. To retain existing values, click **Cancel**.

Managing Trace Files

- To manage trace files, navigate to the Administration tab and click the hyperlink **Trace Files**. As displayed in [Figure 8–9](#), the Trace Files page appears.

Figure 8–9 Trace Files Page



- To view a trace file, select the trace file and click **View**.

Note: When you view the trace file online, the View button truncates trace files that contain more than 10,000 lines. The Download button does not truncate the trace file.

- To download the trace file, click **Download**.
- To delete the trace file, click **Delete**.

Consolidator Components for Tracing

The Consolidator has been categorized into the following five components for tracing. Based on the requirement, you can customize these components.

MGP

This refers to the MGP process except that before an MGP Cycle ID is available, all the logs are covered by the parameter GLOBAL. If the TRACE_DESTINATION is set to the value TEXTFILE, all the generated logs are recorded in a log file named MGP_<cycle_id>.log under the following directory.

<Mobile_Server_Home>\Mobile\Server\ConsLog\MGP

Note: The log file can be viewed directly from the Sync Manager.

MGPAPPLY

This refers to the APPLY phase in the MGP process. However, between the beginning of the APPLY phase till the availability of the MGP Client ID, the logs are covered by the component MGP. If the TRACE_DESTINATION is set to the value TEXTFILE, all the generated logs are recorded in a log file named MGPAPPLY_<client_id>_<cycle_id>.log under the following directory.

<Mobile_Server_Home>\Mobile\Server\ConsLog\MGPAPPLY

Note: The log file can be viewed directly from the Sync Manager.

MGPCOMPOSE

This refers to the COMPOSE phase in the MGP process. Similar to the MGPAPPLY phase where the Client ID is not yet available, the logs are covered by the component MGP. If the TRACE_DESTINATION parameter is set to the value TEXTFILE, all the logs generated are recorded in a log file named MGPCOMPOSE_<client_id>_<cycle_id>.log under the following directory.

<Mobile_Server_Home>\Mobile\Server\ConsLog\MGPCOMPOSE

Note: The log file can be viewed directly from the Sync Manager.

SYNC

This refers to the server side synchronization process. When a Sync session ID is not yet available, the logs will be taken care by component GLOBAL. If the TRACE_DESTINATION parameter is set to the value TEXTFILE, all the logs generated are recorded in a log file named SYNC_<cycle_id>.log under the following directory.

<Mobile_Server_Home>\Mobile\Server\ConsLog\SYNC

When the Client ID becomes available, the file is renamed to SYNC_<client_id>_<cycle_id>.log.

Note: The log file can be viewed directly from the Sync Manager.

GLOBAL

This component logs tracing messages that are not specific to any of the above listed components. This component also includes logs that are generated during the execution of a Consolidator API program. If the TRACE_DESTINATION parameter is set to the value TEXTFILE, all the logs that are generated are recorded in a log file named GLOBAL_<file_number>.log under the following directory.

<Mobile_Server_Home>\Mobile\Server\ConsLog\Global

Note: The <file_number> is used for file pooling.

Tracing Levels and Types

This section describes tracing levels and types that are used to set trace settings.

TRACE_LEVEL

This parameter controls the logging output. Before logging a message, Consolidator first checks if the component is set to a certain level. It then checks the TRACE_TYPE and USER settings. Trace types that are insensitive to the trace level are exceptional. In such cases, Consolidator ignores the trace level unless this parameter is set to OFF.

Trace Levels are ordered and have cascading effects. For example, the mandatory level is higher than the warning level. When a component is set up with a mandatory level, it generates mandatory logs only. But when the component is set up with a warning level, it generates mandatory and warning logs.

-
- **OFF:** This parameter turns off the Tracing feature for the component.
 - **MANDATORY:** This parameter logs mandatory messages only. For example, Program Exceptions. Note that regardless of component settings, exceptions are logged in the `err.log` file, which is located in the `ConsLog` directory.
 - **WARNING:** This parameter logs warning (and the above mandatory level) messages. For example, Program Exceptions that users can ignore, messages that the program wants to warn the users with, and so on.
 - **NORMAL:** This parameter logs normal messages that the user should be informed about. It also logs messages with the above levels.
 - **INFO:** This parameter logs information messages. For example, Timing and messages with the above levels.
 - **CONFIG:** This parameter logs configuration messages and messages with the above levels.
 - **FINEST:** This parameter sets the finest level. This level can be used by developers for development purposes only.
 - **ALL:** This parameter logs all messages according to the other settings, i.e., Trace Type and Users.

Trace_Type

This parameter sets the message type that must be traced. The following parameters can be used to set the trace type parameter.

- **SQL:** This parameter logs SQL-related messages only. For example, SQL statements. Note that the `TRACE_TYPE=SQL` parameter is not trace level sensitive. This option prints all SQL-related messages with the trace level set to any level other than **OFF**.
- **TIMING:** This parameter logs timing information only. You must note that the `TIMING_TYPE` parameter is trace level sensitive. When specifying MGP Cycle time and Synchronization time, choose the `TRACE_LEVEL=INFO` parameter.
- **DATA:** This parameter logs data only. Note that this type is not trace level sensitive. This option prints all data with the trace level set to any level other than **OFF**.
- **RESUME:** Messages dealing with Reliable Transport have a **RESUME** trace type. Components set with this parameter type can log messages associated with Reliable Transport. Note that this trace type is not trace level sensitive. Therefore, choosing this type prints all the **RESUME** messages with the trace level set to any level other than **OFF**.
- **FUNCTION:** This parameter displays the program flow by logging methods' entry, exit or invoke. When this trace type is turned on, `ConsLogger` logs the names of methods that have been called. For long methods, it logs the methods entry and exit; a simple invoke log otherwise. Note that this type is not trace level sensitive. Therefore, choosing this type prints all the **FUNCTION** messages with the trace level set to any other than **OFF**.
- **GENERAL:** This parameter logs messages that do not belong to any of the above types. Note that this type is Trace level sensitive.
- **All:** This parameter generates logs of all trace types.

Users

This parameter is used to limit the trace logging for certain users only. This parameter only affects user-sensitive logs; therefore, if a trace log is not specific to a user, it is logged irrespective of the trace setting in the trace users parameter. When this parameter is not set with any Consolidator Client ID, the program does not check further and log any messages. Otherwise, it checks if the log belongs to one of the Trace users and generates the log only if it does. Note that all invalid Consolidator Clients that are entered are ignored.

8.6 Browsing the Sync Repository

The Repository tab describes how to look up user information, publications, publication items, and the transaction queues that contain in, out, and error queue transactions. This section contains the following topics:

- [Section 8.6.1, "Viewing User Information"](#)
- [Section 8.6.2, "Viewing Publications"](#)
- [Section 8.6.3, "Viewing Publication Items"](#)
- [Section 8.6.4, "Viewing Publication Queues"](#)

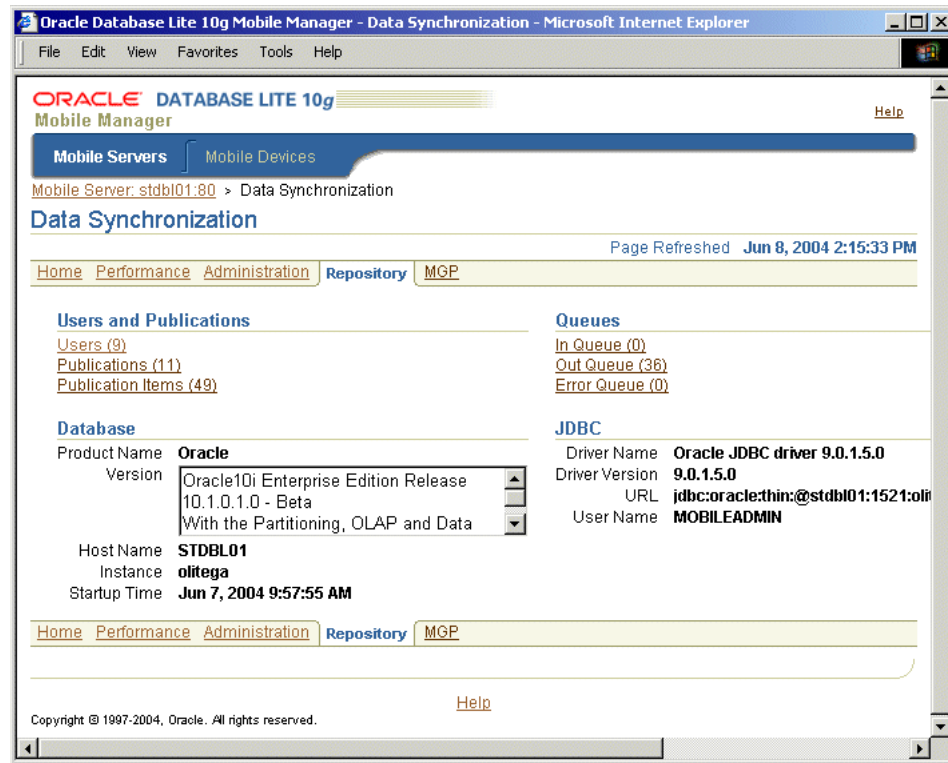
8.6.1 Viewing User Information

The Users page enables you to check application subscriptions, publication items, parameters, SQL scripts, java resources, sequences, and Consp perf performance analysis.

1. To view information about existing users, click the **Repository** tab on the Data Synchronization home page.

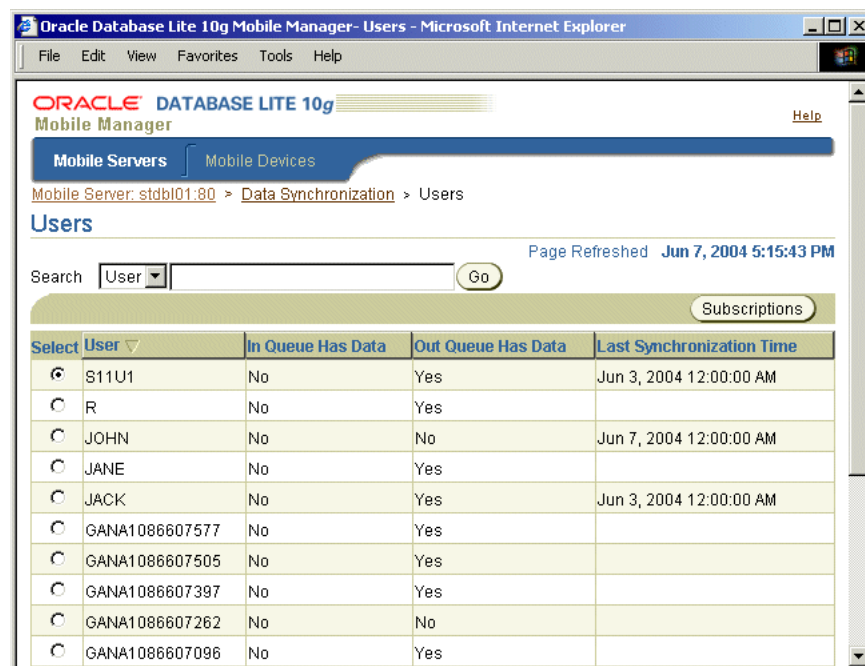
As displayed in [Figure 8–10](#), the Repository tab appears.

Figure 8–10 Repository Tab



- To view information about existing users, click **Users** under the Users and Publications section. As described in Figure 8–11, the Users page appears.

Figure 8–11 Users Page

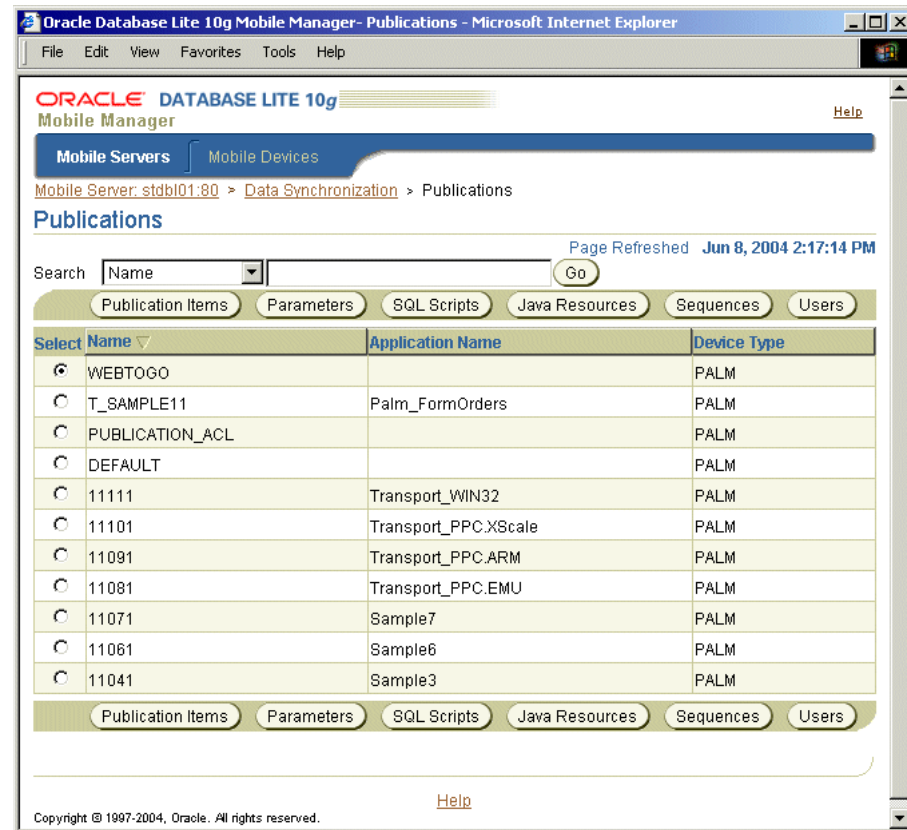


3. To view the application subscriptions of individual users, click **Subscriptions**. The Subscriptions page displays existing subscriptions and controls to view publication items, parameters, SQL scripts, java resources, sequences, and Conspert performance analysis.

8.6.2 Viewing Publications

To view publications, click **Publications** under the Users and Publications section. As displayed in [Figure 8-12](#), the Publications page appears.

Figure 8-12 Publications Page



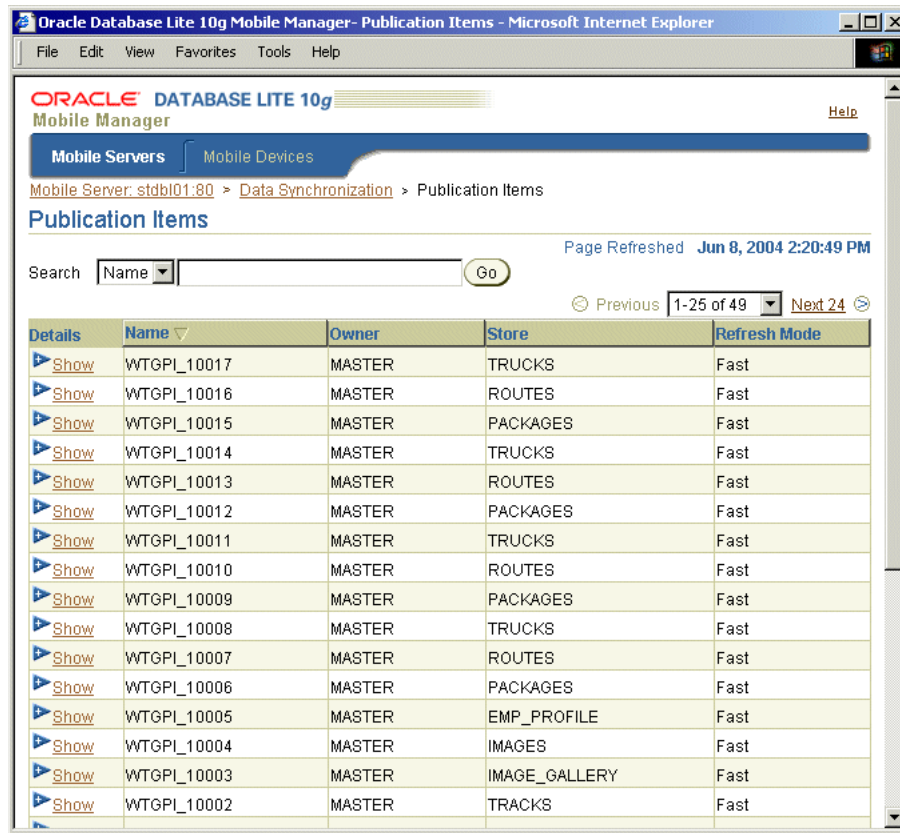
The Publications page displays controls to view publication items, parameters, SQL scripts, java resources, sequences, and users.

8.6.3 Viewing Publication Items

To view publication items, click **Publication Items** under the Users and Publications section.

As displayed [Figure 8-13](#), the Publication Items page appears.

Figure 8–13 Publication Items Page



To display details for a publication item, click **Show**.

8.6.4 Viewing Publication Queues

The Out Queue is organized by subscriptions. A subscription is based on users and publications. The In Queue and Error Queue are organized by transactions. The Repository tab contains controls to view these queues. To view transactions that are listed in queues, click the required hyperlink under the Queues section. For example, to view transactions that are listed in the Out Queue, click **Out Queue**.

Figure 8–14 displays the Out Queue Publications page.

Figure 8–14 Out Queue Publications Page

Oracle Database Lite 10g Mobile Manager - Out Queue Publications - Microsoft Internet Explorer

ORACLE DATABASE LITE 10g Mobile Manager

Mobile Servers Mobile Devices

Mobile Server: stdbl01:80 > Data Synchronization > Out Queue Publications

Out Queue Publications

Page Refreshed Jun 8, 2004 2:22:04 PM

Search User Go

Publication Items

Previous 1-25 of 36 Next 11

Select	User	Publication	Complete Refresh Requested
<input checked="" type="radio"/>	SYSTEM	PUBLICATION_ACL	No
<input type="radio"/>	SYSTEM	WEBTOGO	No
<input type="radio"/>	SYSTEM	DEFAULT	Yes
<input type="radio"/>	S11U1	PUBLICATION_ACL	No
<input type="radio"/>	S11U1	DEFAULT	No
<input type="radio"/>	S11U1	T_SAMPLE11	No
<input type="radio"/>	S11U1	WEBTOGO	No
<input type="radio"/>	R	DEFAULT	Yes
<input type="radio"/>	R	WEBTOGO	No
<input type="radio"/>	R	PUBLICATION_ACL	No
<input type="radio"/>	JANE	DEFAULT	Yes
<input type="radio"/>	JANE	PUBLICATION_ACL	No
<input type="radio"/>	JANE	WEBTOGO	No
<input type="radio"/>	JANE	11041	No
<input type="radio"/>	JIANF	11061	No

To view the In Queue and Error Queue transactions, click the corresponding hyperlink on the Repository tab.

8.7 Scheduling MGP Cycles to Run Inside the Mobile Server

MGP is a background process that uses one or more Java threads to apply transactions that are listed in the In Queue to the back end Oracle database. It then composes server side changes for all clients into transactions and places them in the Out Queues. This process is called an Apply/Compose MGP cycle. In an Apply Only cycle, the Compose phase is omitted. As the Mobile Server administrator, you can schedule MGP cycles using the Job Scheduler. The MGP tab contains a hyperlink to access the Job Scheduler. Based on the Job Engine's status, this link appears as Job Scheduler(Up) or Job Scheduler(Down).

The Job Scheduler is the preferred way to run MGP. In this way, MGP runs threads in the Mobile Server.

Listed below are some advantages of running MGP.

- MGP and Sync service can share resources.
- MGP cycles can be managed and monitored by the Sync Manager.
- MGP cycles can be scheduled to run at any specified time.

There is a default MGP job called MGP_DEFAULT, that starts every one minute. Note that there can be only one MGP cycle at any given time. Hence, when an MGP cycle is

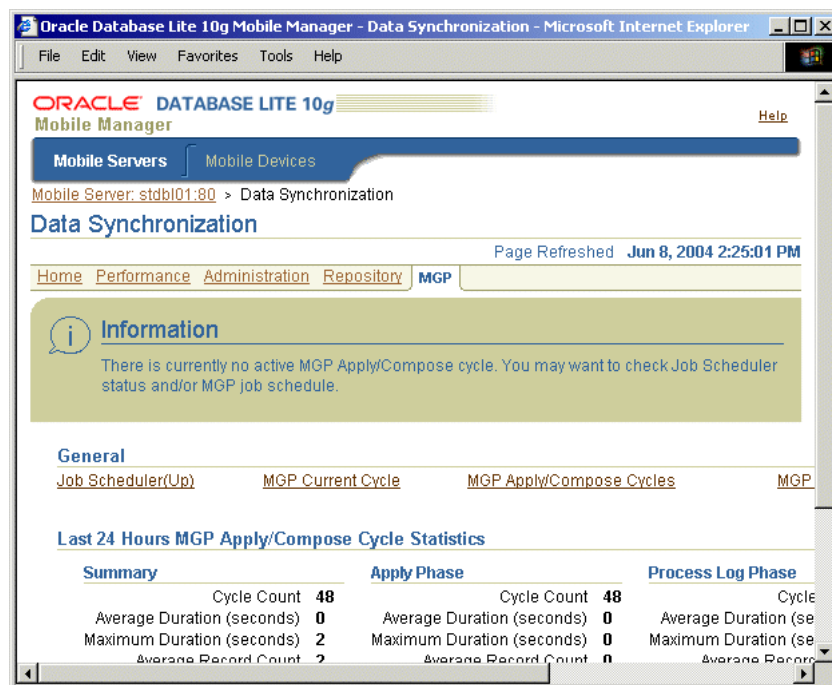
running, a new MGP job cannot be started, even when it is due at a pre-scheduled time.

Using the Synchronization Manager, the Mobile Server administrator can view the current status of the MGP cycle. For instance, the administrator can check if the apply or compose cycle is running when the MGP cycle is in progress. Upon completion of the apply or compose cycle, the corresponding cycle details are stored in Cycle History if the MGP_HISTORY instance parameter is set to the default value TRUE and the cycle is not empty. This status denotes that there are client or server changes processed within an MGP cycle.

The MGP tab describes how to monitor MGP cycles, manage the cycle history, and view statistics of the Apply/Compose cycle.

Figure 8–15 displays the MGP page.

Figure 8–15 MGP Page



To view the current MGP cycle, history cycles or more statistics, click the corresponding hyperlink displayed under the General section.

8.8 Running the Message Generator and Processor (MGP) from the Command Line

Starting with the Oracle Database Lite 10g Release 1 (10.1), the Job Scheduler is the preferred way to run MGP. However, MGP can still be invoked from the command line and be run as a separate OS process. This section describes the functions of the MGP and describes how to configure the MGP process. Topics include:

- [Section 8.8.1, "Overview"](#)
- [Section 8.8.2, "MGP Cycles"](#)
- [Section 8.8.3, "Configuring MGP"](#)

8.8.1 Overview

MGP is a multi-threaded process. As an administrator, you can configure the number of threads with the parameter `MAX_THREADS` in the **CONSOLIDATOR** section of the Mobile Server configuration file **webtogo.ora**. This parameter specifies the number of threads spawned within the MGP process.

For more information, see [Appendix B, "Mobile Server Configuration Parameters"](#).

8.8.2 MGP Cycles

MGP is a cyclical process. An MGP cycle is a large transaction where data is applied to and is composed from the Oracle database server for all users. There are two sub processes within one MGP cycle: one for `APPLY` and one for `COMPOSE`. Using the `DELAY` parameter, you can specify the delay duration between cycles in seconds.

For example, if ten users synchronize their data with the Mobile Server, MGP starts the next cycle after a specified delay in seconds, applying and composing data for these ten users. After this cycle, MGP pauses for the specified number of `DELAY` seconds and starts another cycle.

In certain situations, it is recommended that you shut down the MGP process completely (for example, to release memory) and restart. You can do this by specifying the total number of cycles that you want MGP to execute, before the MGP process stops.

8.8.3 Configuring MGP

The length of a given MGP cycle depends on the data that is being synchronized and on other constraints. As a result, users are not provided any control over MGP cycle time. Users can, however, configure the delay duration between cycles and the total number of cycles between restarts. The following section provides the requisite syntax to set the delay and restart parameters. Users can set these parameters from the command line or in the file **mgp.bat**.

Syntax

```
mgp <delay in seconds between cycles> <number of cycles between
restart> <username> <password>
```

Example

```
mgp 60 10 mobileadmin manager
```

In this example, the delay duration between cycles is 60 seconds and the total number of cycles between MGP restarts is 10.

Setting Parameters Manually on Solaris and Windows

As the Mobile Server administrator, you can set the MGP configuration parameters manually in the file **mgp** (on Solaris) and in the file **mgp.bat** (on Windows). This file resides in the following directory, depending on the platform where the Mobile Server is running.

On Solaris

```
<ORACLE_HOME>/mobile/server/bin
```

On Windows

```
<ORACLE_HOME>/mobile/server/bin
```

Replace the <ORACLE_HOME> variable with the name of your Oracle home directory. Since MGP is multi-threaded, you can configure the number of threads by setting the MAX_THREADS parameter (static only).

Configuring MGP Threads

As the Mobile Server administrator, you can set the parameters SLEEP_TIME and MAX_TIME in the file **webtogo.ora**. The parameter SLEEP_TIME refers to the duration between individual client compose cycles. MGP_DELAY is the duration between each MGP cycle, after all the clients have been composed.

To configure the duration of a thread in sleep mode, set the parameter SLEEP_TIME. When you set this parameter, it applies the COMPOSE sub process. For example, situations may occur where the Oracle database server is busy and composing requires significant CPU, RAM, and I/O resources. In this case, you can provide some relief to the server by setting the MGP threads to sleep mode for a minimum duration.

Example

```
SLEEP_TIME = 10000
```

```
MAX_THREADS = 3
```

In the above example, you can specify the duration between client processes in milliseconds. The parameter MAX_THREADS specifies that the number of threads spawned within the MGP process is 3.

8.9 Monitoring Synchronization Using SQL Scripts

To monitor mobile application status during synchronization, you may use any tool to check for pertinent information in the applicable tables, or you may use SQL scripts to retrieve the desired information.

To help you monitor progress of the synchronization process, the following sections present examples of various SQL scripts that you may use to retrieve different types of information. Topics include:

- [Section 8.9.1, "Shared Maps"](#)
- [Section 8.9.2, "Synchronization Times for All Clients"](#)
- [Section 8.9.3, "Failed Transactions for all Clients"](#)
- [Section 8.9.4, "Completely Refreshed Publication Items for all Clients"](#)
- [Section 8.9.5, "Publications Flagged for Complete Refresh for All Clients"](#)
- [Section 8.9.6, "Clients and Publication where Subscription Parameters are Not Set"](#)
- [Section 8.9.7, "Record Counts for Map-based Publication Item by Client"](#)
- [Section 8.9.8, "Record Count for Map-based Publication Items by Store"](#)
- [Section 8.9.9, "All Client Sequence Partitions and Sequence Values"](#)
- [Section 8.9.10, "All Publication Item Indexes"](#)

8.9.1 Shared Maps

It is very common for publications to contain publication items that are used specifically for "lookup" purposes. In other words, the server may change these snapshots but the client would never update them directly. Furthermore, many users often share the data in this type of snapshot. For example, there could be a publication

item called “zip_codes” that is subscribed to by all mobile users. The main function of Shared Maps is to improve scalability for this type of publication item by allowing users to share record state information and, thus, reduce the size of the resulting replication map tables.

Shared Maps can also be used with updatable snapshots if the developer is willing to implement their own conflict detection and resolution logic.

8.9.2 Synchronization Times for All Clients

Using the following script, you can check the latest successful synchronization times for all clients by retrieving such information from the `all_clients` table.

```
select client, lastrefresh_starttime, lastrefresh_endtime
from cv$all_clients
order by client
/
```

8.9.3 Failed Transactions for all Clients

Using the following script, you can retrieve a list of failed transactions for all clients from the `all_errors` table.

```
select client, transaction_id, item_name, message_text
from cv$all_errors
where message_text is not null
order by client, transaction_id
/
```

8.9.4 Completely Refreshed Publication Items for all Clients

Using the following SQL script, you can retrieve a list of all publication items for all clients which were completely refreshed during the last synchronization process.

```
select clientid, publication_item
from c$complete_refresh_log
order by clientid, publication_item
/
```

8.9.5 Publications Flagged for Complete Refresh for All Clients

Using the following SQL script, you can retrieve a list of publications for all clients that are flagged for a complete refresh during the next synchronization process.

```
select clientid, template as publication
from c$all_subscriptions
where crr = 'Y'
/
```

8.9.6 Clients and Publication where Subscription Parameters are Not Set

Using the following SQL script, you can retrieve a list of clients and their publications where the subscription parameters have not been set.

```
select client, name as publication, param_name, param_value
from cv$all_subscription_params
where param_value is null
```

```
order by client, name
/
```

8.9.7 Record Counts for Map-based Publication Item by Client

Using the following script, you can retrieve record counts for all clients in queues for map-based publication items, that are grouped by clients.

```
select clid$$cs as client, count(*) as "RECORD COUNT"
from c$in_messages
group by clid$$cs
/
```

8.9.8 Record Count for Map-based Publication Items by Store

Using the following SQL script, you can retrieve record counts for all client in-queues for map-based publication items, that are grouped by store.

```
select clid$$cs as client, tranid$$ as transaction_id, store as item_name,
count(*) as "RECORD COUNT"
from c$in_messages
group by clid$$cs, tranid$$, store
/
```

8.9.9 All Client Sequence Partitions and Sequence Values

Using the following SQL script, you can retrieve a list of all client sequence partitions and current sequence values.

```
select clientid, name, curr_val, incr
from c$all_sequence_partitions
order by clientid, name
/
```

8.9.10 All Publication Item Indexes

Using the following SQL script, you can retrieve a list of all publication item indexes.

```
select publication as NAME, publication_item, conflict_rule as "INDEX_TYPE",
columns
from c$all_indexes
order by publication, publication_item
/
```

Job Scheduler

This document enables the Mobile Server administrator to manage the job engine and job schedules through the web. Topics include:

- [Section 9.1, "Overview"](#)
- [Section 9.2, "Managing the Job Engine"](#)
- [Section 9.3, "Administering Scheduled Jobs"](#)

9.1 Overview

The Oracle 10g Job Scheduler provides an Application Programming Interface (API) and a graphical user interface for scheduling and running jobs using a job engine. It is a generic component which enables apply and compose functions for MGP, device manager jobs, and custom jobs.

The Job Scheduler can be used from two interface layers, also known as the inner and outer layers. The inner layer is the API which enables application developers to define jobs, submit jobs, and manage jobs programmatically. The outer layer is a GUI which enables administrators to manage the job engine, and manage job schedules on the web.

The API comprises some methods of the class `oracle.lite.sync.ConsolidatorManager` and other supporting classes such as `Job`, `Schedule`, `ExecutionResult` and `ExecutionLog` in the `oracle.lite.sync.job` package. Application developers can schedule jobs based on a pre-determined time and interval. For example, jobs can be scheduled to run repeatedly for a specified duration on any specified day or days of the week or month. Administrators can schedule jobs to run repeatedly for a specified number of months, weeks or specified days of the month or week. Using the class `ConsolidatorManager`, application developers can register or de-register a job class, create, drop, enable or disable a job, search, and delete a job execution log. For more information, refer the *Consolidator Admin API Java Doc*.

The Job Scheduler home page enables administrators to operate the Job Engine using the following tabs.

- [Home](#)
- [Administration](#)

Home

The Home tab enables Mobile Server administrators to start up or shut down the job engine, check registered alerts, view active jobs, and access the Job History page. The

Home tab displays the status of the job engine. If the job engine encounters any exceptions, the Home tab displays the stack trace of the last exception.

The Alerts table displays severity of an exception as a critical alert under the Severity column. Job Execution Failures are another alert type and are displayed as warning alerts.

The Job History page contains search criteria to search, sort, and manage the job history list. Administrators can search the Job History page based on job properties such as name, class name, result or a specific date and time.

Administration

The Administration tab enables administrators to create a new job and edit existing jobs. It provides administrative functions to enable jobs, disable jobs, and delete jobs. The Scheduled Jobs section displays a list of jobs that are scheduled in the job engine.

9.2 Managing the Job Engine

The Job Scheduler home page enables administrators to start the job scheduler and check for alerts that are registered in the job engine. Using this page, administrators can manage active jobs and the job history list. The sections below enable administrators to accomplish the following tasks. Topics include:

- [Section 9.2.1, "Starting the Job Scheduler"](#)
- [Section 9.2.2, "Checking Job Scheduler Alerts"](#)
- [Section 9.2.3, "Managing Active Jobs"](#)
- [Section 9.2.4, "Managing the Job History List"](#)

9.2.1 Starting the Job Scheduler

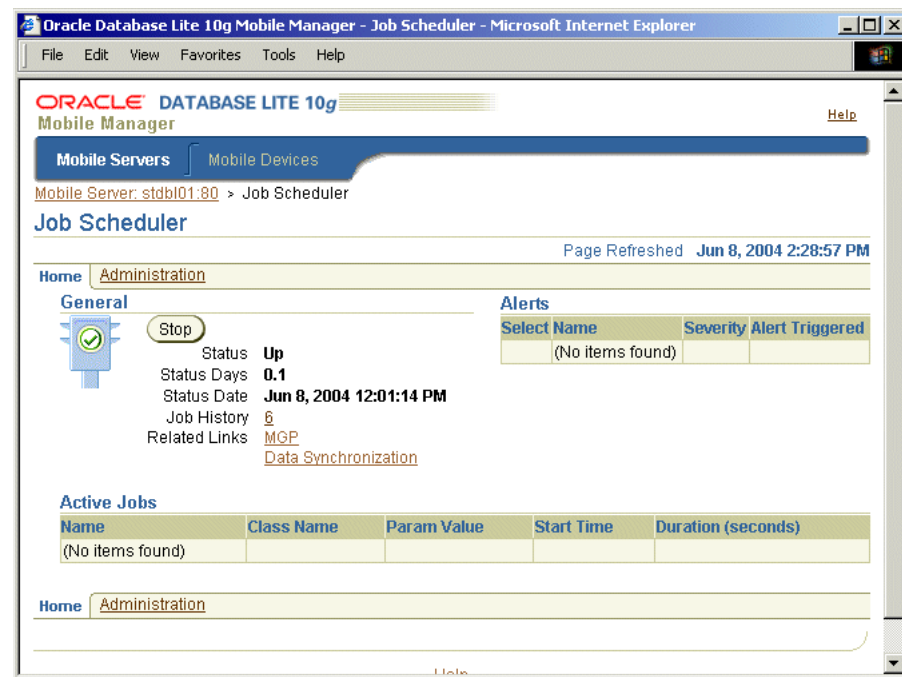
To start the Job Scheduler, navigate to the Job Scheduler home page as follows:

1. Login to the Mobile Server using the appropriate user name and password.
2. Locate the Mobile Server components table, and click **Job Scheduler**. The Job Scheduler home page appears.

The Job Scheduler's default status is "Up". The following image displays the Job Scheduler's default status.

[Figure 9–1](#) displays the Job Scheduler's default status on the Job Scheduler home page.

Figure 9–1 Job Scheduler Home Page

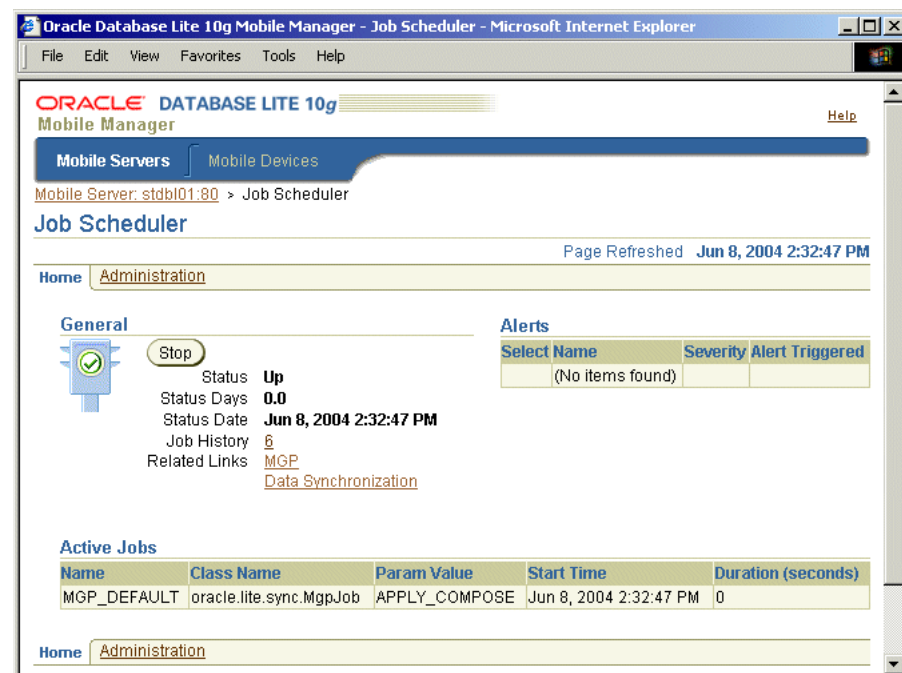


3. To start the Job Scheduler, click **Start**.

At this stage, the "Start" button is replaced by the "Stop" button. The following image displays that the Job Scheduler is up and running.

Figure 9–2 displays the Job Scheduler which is currently running.

Figure 9–2 Starting the Job Scheduler



9.2.1.1 Stopping the Job Scheduler

To stop the Job Scheduler, click **Stop**. The Mobile Server displays a warning message that seeks your confirmation to stop the Job Scheduler. Click **Yes**. You will be returned to the Job Scheduler home page.

9.2.2 Checking Job Scheduler Alerts

The Job Scheduler home page enables administrators to check alerts that are registered in the job engine. To check alerts, locate the "Alerts" table and select the alert that you need to view under the **Select** column. Click **Check**. The Mobile Server displays a list of failed jobs on the Job History page.

Figure 9–3 displays the Job History page.

Figure 9–3 Job History Page

Job History - Microsoft Internet Explorer

ORACLE Enterprise Manager

Mobile Servers

Mobile Server: mhzhou-pr.us.oracle.com:80 > Job Scheduler > Job History

Job History

Page Refreshed Oct 13, 2003 11:29:49 PM

Search

Job Properties

Name:

Class Name:

Result: FAILURE

From

Date: 6/24/03

Time: 2:00 PM

Time Zone: Pacific Standard Time

To

Date: 10/13/03

Time: 11:25 PM

Go Delete All

Results

Previous 1-10 of 1105 Next 10

Select	ID	Name	Class Name	Result	Finish Time	Duration (seconds)	Message
<input checked="" type="radio"/>	6715	Hello_3	oracle.lite.sync.HelloJob	FAILURE	10/13/03 11:23 PM		0 Hello world! Job parameter is "null". java.lang.Exception: Less fortunate: (at oracle.lite.sync.HelloJob.execute (HelloJob.java:51) at oracle.lite.job.JobThread.run (JobThread.java:59)
<input type="radio"/>	6713	Hello_3	oracle.lite.sync.HelloJob	FAILURE	10/13/03 11:22 PM		0 Hello world! Job parameter is "null". java.lang.Exception: Less fortunate: (at

To delete an alert, locate the "Alerts" table on the Job Scheduler home page. Select the alert and click **Delete**.

9.2.3 Managing Active Jobs

The Active Jobs table on the Job Scheduler home page contains information such as job name, class name, parameter value, job start time, and duration. To terminate an active job, click **Kill**. The Job Scheduler displays a warning message that seeks your confirmation to terminate the active job. Click **Yes**. You will be returned to the Job Scheduler home page.

9.2.4 Managing the Job History List

The Job Scheduler home page displays the total number of jobs that are registered in the job history list. Using this page, the administrator can search, sort, and manage the job history list, which is based on job properties.

To display the job history list, click the **number** hyperlink which is displayed against Job History. For example, click the number which is displayed as a hyperlink against Job History. The Job Scheduler displays the Job History page.

To search the job history list, enter your search criteria based on job properties such as job name, class name, date, and time in the corresponding fields. Click **Go**. Based on your search criteria, the Job History page displays job history details under the **Results** section.

To sort job history details, click the required **Header Title**. For example, to sort job history details by name, click **Name** in the header title region.

To delete a job, select the job and click **Delete**.

The Job Scheduler enables you to delete all job history entries that match your search criteria. To delete job history entries that match your search criteria, click **Delete All**.

9.3 Administering Scheduled Jobs

The Administration tab enables administrators to create a new job, edit existing jobs, and enable, disable or delete jobs. The sections below enable administrators to accomplish the following tasks. Topics include:

- [Section 9.3.1, "Creating a New Job"](#)
- [Section 9.3.2, "Editing Existing Jobs"](#)
- [Section 9.3.3, "Enabling Jobs"](#)
- [Section 9.3.4, "Disabling Jobs"](#)
- [Section 9.3.5, "Deleting Jobs"](#)
- [Section 9.3.6, "Default Jobs"](#)

9.3.1 Creating a New Job

To create a new job, navigate to the **Administration** tab and click **Create A New Job**. The Create a New Job page appears. The following images display the Create a New Job page.

[Figure 9-4](#) displays the top section of the Create a New Job page.

Figure 9–4 Create a New Job - Top Section

Oracle Database Lite 10g Mobile Manager - Create A New Job - Microsoft Internet Explorer

File Edit View Favorites Tools Help

ORACLE DATABASE LITE 10g Mobile Manager

Mobile Servers Mobile Devices

Mobile Server: stdbl01:80 > Job Scheduler > Create A New Job

Create A New Job

Page Refreshed Jun 8, 2004 2:47:44 PM

Job

General

Job Name

☐ Enabled

☐ Save to Job History

Job Class

☒ MGP

Apply/Compose Mode

☐ Custom

Class Name

Parameter Value

Schedule

Start

☒ Immediately

Expiration

☒ Never

Time Zone India Standard Time

Figure 9–5 displays the bottom section of the Create a New Job page.

Figure 9–5 Create a New Job - Bottom Section

Oracle Database Lite 10g Mobile Manager - Create A New Job - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Start

☒ Immediately

☐ Later

Date

Example: 10/31/03

Time ☐ AM ☒ PM

Expiration

☒ Never

Expire

☐ Expire

Limit (minutes)

Cancel if not started within time limit

Repeat

☒ One

Time Only

☐ Interval

Frequency (seconds)

☐ Weekly

Frequency (weeks)

Days of Week

☐ Mo ☐ Tu ☐ We ☐ Th ☐ Fr ☐ Sa ☐ Su

☐ Monthly

Frequency (months)

Days of Month

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7

☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12 ☐ 13 ☐ 14

☐ 15 ☐ 16 ☐ 17 ☐ 18 ☐ 19 ☐ 20 ☐ 21

Repeat Until

☒ Indefinite

☐ Custom

Date

Example: 10/31/03

Time ☐ AM ☒ PM

Enter data in the Create a New Job page as described in the following tables.

Table 9–1 describes data that must be entered in the **General** section of the Create a New Job page.

Table 9–1 General Details Description - Job Section

Field	Description	Required
Job Name	Unique Job Name	Yes
Enabled	To enable a job, select this check box.	Optional
Save To Job History	To save a job's execution information in the Job History, select this check box.	Optional

Table 9–2 describes data that must be entered in the **Job Class** section of the Create a New Job page.

Table 9–2 Job Class Details Description - Job Section

Field	Description	Required
MGP	To specify <code>oracle.light.sync.MgpJob</code> as the job class, select this option. Note: If selected, choose the appropriate Apply/Compose Mode from the list displayed.	Optional
Custom	To specify a custom class for the job, select this option. Note: If selected, enter the Custom Class Name and its Parameter Value in the corresponding fields.	Optional

Table 9–3 describes data that must be entered in the **Start** section of the Create a New Job page.

Table 9–3 Start Details Description - Schedule Section

Field	Description	Required
Immediately	To start the job immediately, select this option.	Optional
Later	To start the job at a later time, select this option. Note: If selected, select the appropriate Date and Time from the corresponding fields.	Optional

Table 9–4 describes data that must be entered in the **Expiration** section of the Create a New Job page.

Table 9–4 Expiration Details Description - Schedule Section

Field	Description	Required
Never Expire	To ensure that the chosen job schedule does not expire, select this option.	Optional
Expire	To indicate that the chosen job schedule will expire within a specified time limit, select this option. The Job Scheduler cancels jobs that do not start at the specified time. However, it does not stop jobs that have already started. Note: If selected, enter the job expiry duration in the Limit (minutes) field.	Optional

Table 9–5 describes data that must be entered in the **Repeat** section of the Create a New Job page.

Table 9–5 Repeat Details Description - Repeat Section

Field	Description	Required
One Time Only	To indicate a one time repeat schedule, select this option.	Optional
Interval	To indicate a repeat schedule with an interval, select this option. Note: If selected, enter the interval duration in seconds, in the corresponding field.	Optional
Weekly	To indicate a weekly job repeat schedule, select this option. Note: If selected, choose a number that specifies the weekly repeat frequency from the Frequency (weeks) list. To indicate the repeat schedule on specific days of the week, select the relevant Days of Week boxes.	Optional
Monthly	To indicate a monthly job repeat frequency schedule, select this option. Note: If selected, choose a number specifying the monthly repeat frequency from the Frequency (months) list. To indicate the repeat schedule on specific days of the month, select the relevant Days of Month boxes.	Optional

Table 9–6 describes data that must be entered in the **Repeat Until** section of the Create a New Job page.

Table 9–6 Repeat Until Details Description - Repeat Section

Field	Description	Required
Indefinite	To repeat the job schedule indefinitely, select this option.	Optional
Custom	To specify a custom repeat schedule, select this option. Note: If selected, select the appropriate Date and Time from the corresponding fields.	Optional

To implement the job schedule after specifying changes to the schedule, click **OK**.

To retain or restore previous job schedule values, click **Cancel**.

Note: The calendar does not display the selected date if the java script feature in your browser, any pop up blocking tools or search tools are installed and enabled.

9.3.2 Editing Existing Jobs

To edit existing jobs, click **Edit**. The Job Scheduler displays the Edit Job page which contains controls to edit a job schedule.

Figure 9–6 displays the Edit Jobs page.

Figure 9–6 Edit Jobs Page

The controls to edit an existing job schedule are the same as the controls that are provided to create a new job schedule. For more information on how to edit job schedule information, refer the relevant field description table in the previous section.

To implement the job schedule after specifying changes to the schedule, click **OK**.

To retain or restore previous job schedule values, click **Cancel**.

9.3.3 Enabling Jobs

To enable a job, select the job that you need to enable, and click **Enable**. The **Status** column confirms the changed status.

9.3.4 Disabling Jobs

To disable a job, select the job that you need to disable, and click **Disable**. The **Status** column confirms the changed status.

9.3.5 Deleting Jobs

To delete a job, select the job that you need to delete, and click **Delete**. The Job Scheduler displays a warning message that seeks your confirmation to delete the chosen job. Click **Yes**. You will be returned to the Administration tab.

9.3.6 Default Jobs

The Oracle Database Lite 10g Edition contains default jobs. As a user, you can enable or disable these default jobs and edit or delete them. This edition contains the following default jobs.

- MGP_DEFAULT
- PURGE_HISTORY_DEFAULT

9.3.6.1 MGP_DEFAULT

This section describes the job class and the job parameter value.

Job Name

MGP_DEFAULT

Description

This job invokes the MGP process.

Job Class

`oracle.lite.sync.MgpJob`

Job Parameter Value

APPLY_COMPOSE

The parameter value must be a string of the value APPLY_COMPOSE or APPLY_ONLY. When scheduling or editing this parameter using the Job Scheduler's Edit Jobs page, you can choose the required parameter value from the Apply/Compose Mode list.

9.3.6.2 PURGE_HISTORY_DEFAULT

This section describes the job class, job parameter value and its corresponding description.

Job Name

PURGE_HISTORY_DEFAULT

Description

This job purges old records in history tables.

Job Class

`oracle.lite.sync.PurgeHistoryJob`

Job Parameter Value

History=Sync,MGP,Job;Days=7

The parameter value must consist a string of semicolon(";") separated name-value pairs. In this parameter value, there are two names called 'History' and 'Days'. The parameter name and its value are separated by the sign "=". The parameter name 'History' enables users to specify the required history component that needs to be purged. For example, in the above parameter value, Sync History, MGP History and the Job History is specified for purging. The parameter name 'Days' enables users to specify purging history records that are old by a certain number of days. For example, in the above parameter value, the number of days for purging the Sync History, MGP History and the Job History is specified as 7.

Managing Devices

This document describes how to manage devices. Topics include:

- [Section 10.1, "Overview and Architecture"](#)
- [Section 10.2, "Device Management Client"](#)
- [Section 10.3, "Software Management Client \(SMC\)"](#)
- [Section 10.4, "Device Management Server"](#)

10.1 Overview and Architecture

As enterprises deploy more and more applications on small devices, managing these devices presents a crucial constraint on enterprise IT solutions. The Oracle Database Lite 10g, Device Management (DM) system provides a solution to administer the deployed devices and remotely manage applications and data in these devices.

The Oracle Database Lite 10g Device Management System consists of a number of Device Management Clients (DMC) that interact with a Device Management Server (DMS). The DMC runs on the target device and the DMS runs on top of the Oracle Database Lite Mobile Server. For the DMS to interact with the remote DMC, the Administrator must enable a compatible networking layer. For example, HTTP, Wake On Ring, and so on. The DMS uses the Mobile Server's user model and provisioning functionality to provide a unified device, software and data management features.

The DMC is a collection of executables that perform various DM tasks. The DMC consists of a Device Management Agent (DMA), Software Management Client (SMC), Software Update Checker and a set of Device Management APIs for applications. The DMA handles basic Device Management functionalities such as command execution and the SMC handles client software management.

10.1.1 Functionality

Oracle Database Lite Device Management supports the following functionality.

1. Remote inspection of client device hardware and operating system settings.
2. Remote inspection and modification of application configuration settings.
3. Client database information retrieval and validation.
4. Client databases synchronization with the Oracle Database.
5. Client device lock down, application removal, application data removal.
6. Client software management.
7. Device configuration files modification. For example, ODBC.INI.

8. Remote database integrity verification.

10.1.2 Terms and Definitions

This section familiarizes you with the terms and definitions used during device management.

Repository

Database schema in which all DMS objects are recorded. DMS provides APIs to access Repository objects. The Device Management Repository is a part of the Mobile Server Repository.

Administrator

Person who administers DMS objects using the Mobile Manager.

Provider

External implementation of a functionality required by the DMS. For example, the DMS depends on the Network Provider to transmit data to devices. If you have a custom network in your environment, you must create the Network Provider class using the JAVA language and register it with the DMS, so that the DMS can transmit messages to remote devices. A Network Provider must implement the `oracle.lite.provider.NetworkProvider` interface.

For a complete definition of this interface, refer the corresponding JAVADOC.

This release of the Oracle Database Lite 10g Device Manager ships with the following network providers.

1. HTTP - DMS uses HTTP protocol to send commands to DMC.
2. RAPI - Remote API (Microsoft® ActiveSync API) Network Provider only supports Pocket PC class devices that are connected directly to the computer which is running the Mobile Server.
3. WOR - Wake on ring based network protocol.
4. SMTP - E-mail based Network Provider.

INF File

An INF file contains software installation instructions. For every application in the Mobile Server, it is desirable to have a unique INF file. For applications that do not have an INF file associated with it, the DMS generates a default INF file.

Developers are highly encouraged to create their own custom INF file for their applications. For more information about the INF File format, refer [Section 10.3, "Software Management Client \(SMC\)"](#).

User

Person who uses a device. ACL (access control lists) are assigned on a user basis. Users can be grouped together for administration purposes.

User Policy

The user policy determines a user's activities. For example, if a user policy setting contains `device.register=true`, the user can automatically register the device with the DMS. If a user's policy reads `device.update=false`, the user does not receive new software updates.

Platform

Platform is a logical operating environment in which the Oracle Database Lite 10g client runs. Each platform has the following set of attributes.

- Name - Unique name that identifies the platform.
- Type - Unique type identifier (For example, WIN32_x86_us) consisting of Operating System Type, CPU type, and Locale.
- INF File name - Installation and configuration file name.
- BOOTSTRAP Commands - Commands to be executed during the platform and device initialization.
- General flags:
 1. Enabled - Indicates if the platform is enabled or not.
 2. Update - Indicates if the platform software is upgradable.
 3. Update apps - Indicates if the application software written for this platform is upgradable.

Listed below are some platforms that are shipped with this release.

1. Oracle Lite WIN32
2. Oracle Lite WEB
3. Oracle Lite PPC2003 ARMV4x

Application

Logical collection of executable code, settings (registry, INI, and so on) and installation actions (creating folder links, executing processes, and so on). Once an application is developed and tested by developers, the Administrator must publish the application to the Mobile Server. To properly install the Software Management Client on the client device, Oracle recommends that the developer also create a custom INF file.

Each application belongs to an Oracle Database Lite platform. To start using an application after publishing, the Administrator must perform the following tasks.

- Assign an Oracle Database Lite platform to the application.
- Assign access rights to users.
- Create publication subscription values to publication items. For more information, refer [Chapter 8, "Managing Synchronization"](#).

Device

Logical representation of a physical device. Only one platform can be installed on a device. Each device has the following set of attributes.

- Name - Unique name to identify the device
- Owner - Owner of the device (User Object)
- Type - Unique type identifier consisting of the Operating System type, CPU type, and Locale. For example, WIN32_x86_US.
- General Flags:
 1. Enabled - Indicates if the Device is enabled or not.
 2. Update - Indicates if the device has the capability for software or data updates.

The software management client installs applications written for the device.

Address

Unique network identifier of a device. The device address must have a corresponding Network Provider associated with it. To transmit data to a device, the DMS uses the Network Provider associated with the address object.

To enable a communication link between the DMS and the DMC, the Administrator must create a proper device address for all devices.

Command

An instruction to the DMC to perform specified actions. These actions can be specified using the OTL scripting language. The first step in creating a custom command is to create an OTL file. Subsequently, you must register the command with the DMS.

The Oracle Database Lite 10g release ships with a set of pre-defined commands and their associated OTL files. A command can be sent to a device in three different ways.

1. Using the registered address of the device.
2. Scheduled to be sent at a later time.
3. Queuing in the device queue. A device pulls these commands later.

Bootstrap

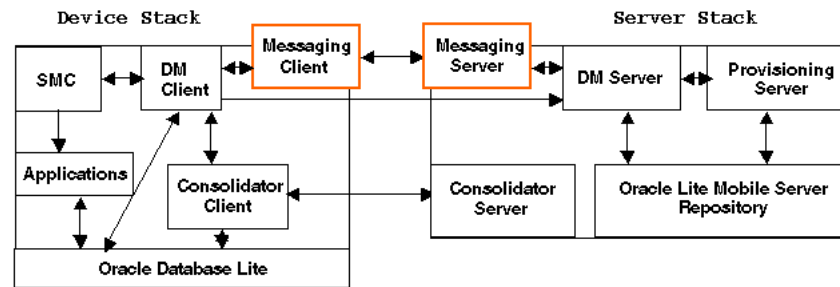
Process of initializing the device so that it can be managed by the DMS. During the bootstrap process, the device registers itself with the DMS. Upon successful registration, the device executes (bootstrap) commands sent by the DMS.

When a device bootstraps with the DMS, it automatically registers its default address with the DMS. For example, a Microsoft Windows based Laptop or Desktop registers its IP Address with the DMS. Whereas, a Pocket PC based device registers RAPI (Active Sync). It is imperative that the Administrator change these default address values to the correct value.

10.1.3 Architecture

As [Figure 10–1](#) displays, the DM System operations are based on the Command and Response paradigm. An Administrator can create device specific commands and send them to the DMC. Upon receiving the commands, the DMC processes them and sends the results to DMS.

The DMS then sends the commands using one of the Network Providers defined by the administrator. The Oracle Database Lite Device Manager ships with a set of Network providers - HTTP, RAPI and Wake-on-Ring. If your company uses any other messaging mechanism, you must implement your own Network Provider implementation and must register it with the DMS.

Figure 10–1 Oracle Database Lite 10g Device Management Architecture

10.1.4 Security

High-level security is one of the main design requisites of the DMS. The main aspects of DM security are summarized below. Topics include:

- [Section 10.1.4.1, "Database Security"](#)
- [Section 10.1.4.2, "Communication Security"](#)
- [Section 10.1.4.3, "Command Execution"](#)
- [Section 10.1.4.4, "Database Synchronization Security"](#)

10.1.4.1 Database Security

The Oracle Database Lite software enforces security between a client, device, and the database. To connect to the database, applications require the User name and Password combination. For a higher degree of security, the entire database can be encrypted.

10.1.4.2 Communication Security

There are two kinds of messages that are sent between the DM Client and the DM Server.

1. Commands sent from the DMS to the DMC.
2. Messages sent from the DMC to the DMS.

The Oracle Database Lite 10g software enforces the communication security process in the following sequence.

1. All the messages exchanged between the DMS and the DMC are encrypted using the AES algorithm.
2. Before the DMC can send messages to the DMS, it must authenticate with the DMS. DMS authentication is based on the HMAC protocol.

10.1.4.3 Command Execution

DM commands are written in the OTL scripting language. The DM provides a sand box style environment for script execution. This prevents a malicious OTL script file from doing irreparable damage to the device.

The OTL scripting engine provides a highly secure environment for command execution. Script files are restricted within the scope of device management functionality and cannot perform any other operation on the client device.

10.1.4.4 Database Synchronization Security

The DM uses the Consolidator client's security features to provide secure database synchronization. To achieve a high-level of security, administrators must configure the DM client to use SSL as the transportation protocol.

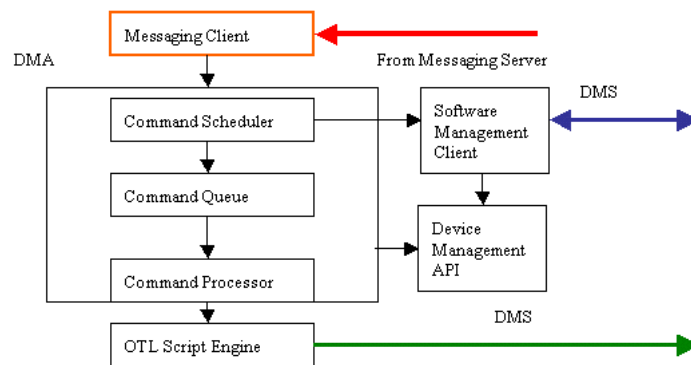
10.2 Device Management Client

When you install the required Mobile Client on a device (Windows Desktop/Laptop or Pocket PC based devices), the corresponding Device Management Client is installed. However, DM features may not be enabled at the time of Mobile Client Installation. To enable the DM feature in the device, it must be bootstrapped with the DMS. The Bootstrap process depends on the device's ability to connect to the Mobile Server, where the DMS is running.

The current implementation of the DMC communicates with the DMS using HTTP (or HTTP over SSL) protocol only. Once the device has been bootstrapped with the DMS, it is ready to accept device commands from the DMS.

Figure 10–2 illustrates the Device Management Client architecture.

Figure 10–2 Device Management Client Architecture



10.2.1 Device Management Agent

The Device Management Agent (DMA) executes commands and transmits results to the DMS. As DM commands are not a fixed set, users can create their own customized commands. All commands are mapped to OTL script pages. For more information about OTL, refer the *Oracle Tag Language Reference*. When a command originator (SMS Client or Wake On Ring Handler) invokes the DMA to execute a command, the DMA searches for the OTL script file represented by the command and executes the script file.

Command execution is queue driven. Therefore, a command sent by the server may not be executed immediately. However, the DMC guarantees command execution and discards duplicate commands.

The following sections describe the bootstrap process, configuration parameters, and the user interface. Topics include:

- [Section 10.2.1.1, "Bootstrap Process"](#)
- [Section 10.2.1.2, "Configuration Parameters"](#)
- [Section 10.2.1.3, "User Interface"](#)

10.2.1.1 Bootstrap Process

All devices must be registered (bootstrapped) with the DMS before it can be managed by the DMS. Bootstrap process involves registering the device with the DMS and executing a set of BOOTSTRAP commands sent by the DMS.

Bootstrapping Windows CE/Pocket PC Based Devices

To bootstrap Windows CE based devices, install Oracle Database Lite on your device and later execute the `dmagent.exe` program which is located in the `\Orace` directory. If the device is not bootstrapped already, the DMAGENT displays the Bootstrap dialog.

To bootstrap the device, enter the appropriate values and press OK. Ensure that your Pocket PC device is connected to the network (using ActiveSync version 3.7 or later) or Ethernet/dialup/wireless connectivity. Oracle strongly recommends using the IP Address instead of the HOST NAME for the Server URL.

Once the bootstrap process is completed, the device executes bootstrap commands received from the server.

Bootstrapping Windows based Desktops/Laptops

If you have installed your Oracle Database Lite client using the `setup.exe` program provided by Oracle Corporation, your device is bootstrapped automatically. However, if you have installed the Oracle Database Lite client using other methods, you must bootstrap your Desktop or Laptop by invoking `dmagent.exe` program before using it.

10.2.1.2 Configuration Parameters

Configuration parameters of the DMA are located in the `devmgr.txt` file, which is located in the `OLITE_HOME` directory. All the Boolean values must be set to 1 (true) or 0 (false). Main configuration items are located under the Section [OMC]. [Table 10–1](#) describes each configuration item.

Table 10–1 Configuration Item Description

Parameter Name	Description
DEVICE_ID	The Device_ID parameter may be pre-configured by the Administrator. If the ID is not pre-defined, the DMS automatically generates an ID during the Bootstrap process.
SERVER_ID	Unique system generated Server ID.
SERVER_URL	HTTP URL of the Device Management Server.
APPS_DIR	Directory in which all applications are installed.
SHOW_UI	Display DM Client's UI (default:1)
MAX_RETRY	Maximum number of attempts to execute a command.
UPDATE_DAY	Day on which the automatic update is scheduled.
UPDATE_TIME	Time at which the automatic update is scheduled.
DISABLE_PROMPT	Disable user confirmation prompting (I/O).
PROMPT_PASSWORD	Prompt for a user's password when DMAGENT is launched (default:1).

REMOTE ACCESS CONFIGURATION PARAMETERS

Remote access (RAS) related configuration parameters are under the section [Network]. These values may not be valid for all devices. [Table 10–2](#) describes remote access related configuration parameters.

Table 10–2 Remote Access Configuration Parameters

Parameter Name	Description
NAME	Remote Access Service Name. For example, USB, WOR, and so on.
DISCONNECT	Disconnect RAS connection when idling (true/false).

To store data that is needed, the DMA relies on the following two sub-directories under the OLITE_HOME directory.

1. APPS\DM - This directory holds all DM commands.
2. USERNAME - Command queue directory.

10.2.1.3 User Interface

The DMA displays the following three dialog boxes.

1. BOOTSTRAP - This dialog box is displayed when the user invokes a DMA on non-bootstrapped devices.
2. Device Management Agent - This is the main dialog. This dialog box is hidden until the SHOW_UI parameter is set to TRUE in the devmgr.txt file. On a Microsoft Windows based desktop or laptop, a tray icon is displayed when the DMA is running. To invoke the DMA user interface, end users can double-click this icon. To invoke the user interface on Pocket PC based devices, users can click the DMAGENT.EXE program, which is available under the following directory.

```
\Oracle\dmagent.exe
```

This user interface enables users to accomplish the following tasks.

1. Check versions of all files.
2. Set or reset DMA options.
3. Password Dialog - The DMA displays this dialog box when invoked by double-clicking the DMAGENT.EXE program. Using this dialog, users can change their password. If the PROMPT_PASSWORD parameter in the devmgr.txt file is set to 0 and a valid password is already available, the DMA does not display this dialog.

10.2.2 Software Update Client

The client UPDATE program named update.exe in Microsoft Windows based devices checks for software updates in the Mobile Server. Software updates include both the Oracle Database Lite system software updates as well as client application updates. Normally, the DMAGENT or MSYNC programs (Oracle Database Lite Synchronization Client) invokes the UPDATE program to check for software updates. However, end users can manually launch the UPDATE program by clicking on the update.exe.

To turn off the automatic update feature, navigate to the Device Properties page and choose 'Yes' from the 'Upgradable' list. This action sets the DISABLE_UPDATE flag in the devmgr.txt file to '1'.

When software updates are available in the Mobile Server, the update program invokes a dialog where end users can choose the application(s) that they wish to install. Users can remove the applications that they do not wish to install immediately. The Administrator can disable the user confirmation dialog by setting the `DISABLE_PROMPT` parameter in the `devmgr.txt` file to '0'.

To install applications, click the 'Install' button. The Update program launches the Software Management Client (SMC).

10.2.3 Software Version Checker

This program displays information such as the current version of the platform, application, and files. To view the platform version, navigate to the Device Management user interface and click on the 'Version' button. Alternatively, users can invoke it by double-clicking on the `olver.exe` program file.

10.3 Software Management Client (SMC)

This chapter describes the Software Management Client (`setup.exe`) in detail and explains how to create a custom INF file for an application. The SMC provides the following functionalities.

- Software Installation
- Software De-install (Un-install)
- Application Information Retrieval

Oracle highly recommends developers to create custom INF files for their applications. It must be named as `setup.inf` and must reside in the root directory of your application. When the Administrator publishes the application to the Mobile Server, the INF file is copied to the repository.

10.3.1 Software Installation

Software Installation involves copying files, updating registry settings, modifying configuration files, creating ODBC DSN, and so on. All these operations are achieved by interpreting the installation configuration file by the SMC. To start the installation process, the SMC sends the application name and current version number to the DMS. The DMS in turn locates the current INF file for such applications. If an update is available, the DMS sends the INF file to the SMC.

A typical software installation comprises the following steps.

1. Software Management Client - Logon to the Oracle Database Lite 10g Mobile Server.
2. Software Management Client - Requests application installation configuration file (INF file).
3. Device Management Server - Sends the INF file to the SMC.
4. Software Management Client - Interprets the INF file.

10.3.2 Installation Configuration (INF) File

The SMC relies on an Installation Configuration file to complete software installation, upgrade, and de-installation. This file contains all the instructions required to install or de-install client software. The Oracle Database Lite Device Management's INF file format is based on XML. The INF file contains a set of ACTIONS and each action may

have multiple items. The SMC parses INF files and performs the necessary actions for each action element in the INF file.

The SMC supports a set of keywords that can be used in any INF file. As [Table 10–3](#) describes, a keyword name starts with a '\$' character and ends with a '\$' symbol.

Table 10–3 Software Management Client Keyword Description

Keyword	Description
\$APP_DIR\$	Application directory of the application
\$APP_NAME\$	Application name
\$OS_DIR\$	Operating system directory
\$OS_TYPE\$	Operating system type. For example, WIN32, WINCE, LINUX, PALM, and so on.
\$OS_VER\$	Operating system version. For example, NT, 95, XP, 3.0, and so on.
\$OS_LANG\$	Language or Location name. For example, EN for English, JP for Japanese.
\$DESKTOP\$	Folder name of the Windows desktop.
\$CPU\$	Device processor type. For example, x86, ARM, XSCALE, MIPS, and so on.
\$HOST_NAME\$	Host name of the client device.
\$USER_NAME\$	User name
\$HTTP_PROXY\$	HTTP Proxy Server name, if any.
\$SERVER_URL\$	Oracle Mobile Server URL.

10.3.2.1 Setup Information

All Software Management actions are enclosed within the `SETUP` XML tag. The `SETUP` consists of a set of `PROPERTIES`, `INITIALIZATION`, `INCLUSION` of other INF files and `INSTALLATION` actions. All the four items must be child elements of the `SETUP` element. A sample INF file is given below.

```
<setup name="Oracle Lite" version="1.0.0.0" type='WIN32_XP_x86'>
<property>... </property>
  <init>...</init>
  <include>...</include>
  <install> ....</install>
</setup>
```

Setup may have the following attributes specified as XML tag attributes.

1. **NAME** - Application name (Mandatory).
2. **VERSION** - Application version number (Mandatory).
3. **PACKAGE** - Package Type. For example, cab - Windows CAB format.

10.3.2.2 Properties

All the `SETUP` properties must be the child element of the `PROPERTY` tag. Setup may have following properties.

- **STORAGE** - Estimated disk (storage) space (in MB) required for an application. For example,


```
<property><storage>5</storage></property>
```

- **MEMORY** - Minimum amount of system memory (in MB) required. For example,

```
<memory>5</memory>
```
- **LOCATION** - Location or directory name of the application. For example,

```
<location default='olite' type='WIN32'></location>
```
- **PORT** - This is an optional value and is only used when installing Oracle Database Lite clients.

- **PROMPT** - All the pre-setup user prompt actions such as terminating running instances of applications are placed in this section. For example,

```
<prompt><item type='WINCE' file='olobj40.dll' />Would you like to
terminate Oracle Lite Application?</prompt>
```

In the above example, **SETUP** prompts user confirmation when it detects an application using the **OLOBJ40.DLL** running in the device. For example,

```
<setup name="Oracle Lite" version="1.0.0.0" type='WIN32_XP_x86'>
<property>
    <storage>4</storage>
    <memory>12</memory>
    <port>80</port>
    <location>d:\tmp\<a</location>
    <prompt>
        <item'>Would you like to install App1?</item>
        <item file='olobj40.dll'>Would you like to close Oracle
Lite Applications></item>
    </prompt>
</property>
```

10.3.2.3 Initialization

Initialization includes setting environment variables.

```
<init> <item type='WIN32' name='DMC_DIR'>$APP_DIR$/bin</item> </init>
```

10.3.2.4 Including Other INF Files

The following syntax allows an INF file to include other INF files.

```
<include>/dmc/common/webtogo.inf</include>
```

The value of this tag can be an application name or a fully qualified INF file name. If the value is an application name, the DMS includes the INF file of the application.

10.3.2.5 INSTALL Element

This section lists all the installation steps necessary to perform Software Installation. Each of the steps (actions) must correspond to another child entry or tag. Each action element has a set of **ITEMS** and two optional caption strings. The caption string is displayed on the SMC's user interface. For example,

```
<action msg_i='Creating directories' msg_u='Removing
directories'>directory</action>
```

When the SMC interprets the above tag, it looks for a child element by the name **DIRECTORY** and processes all the child items of this element. At this stage, the Device Manager UI indicates that directories are being created.

Table 10–4 describes INSTALL actions that are supported by the SMC.

Table 10–4 *INSTALL Actions Supported by the SMC*

Action	Description
directory	Lists all directories to be created.
file	Lists all the files to be copied.
env	Lists all the environment variables to be added to the Operating System.
registry	Registry keys and values to be added to the Windows Registry.
odbc	ODBC driver and DSN to be created.
java	JRE to be installed in the computer.
link	Folder links to be created. For example, desktop, menu, and so on.
ini	INI (configuration files) to be updated.
registry	DLL (or COM objects) to be registered with Windows.
execute	Executable files to be launched during the installation process.
finish	Installation completion messages.

DIRECTORY

The directory named ACTION contains names of all the directories to be created or removed during the installation or removal process. Entries in this section are fully qualified directory names. For example,

```
<directory>
  <item>$APP_DIR$\olddb40</item>
  <item>$APP_DIR$\crm</item>
</directory>
```

The SMC creates OLDB40 and CRM directories in the ROOT of the application directory.

FILE

This section lists all the files to be copied or removed during the software installation or removal process. Each item contains a target file name and source file name. For example,

```
<file>
  <item>
    <src> win32/crm/crm.dll </src>
    <des>$APP_DIR$\crm\crm.dll </des>
  </item>
</file>
```

ENV

This section contains all environment variables to be added to the Operating System.

```
<env>
  <item name='PATH'>$APP_DIR$\WEBTOGO</item>
</env >
```

The above example adds the APPLICATION_ROOT\WEBTOGO directory to the PATH environment variable.

REGISTRY

This section modifies or removes Windows Registry values. All the entries in this section must be a fully qualified registry key name. Sub key names and values must be specified as a sub section. For example,

```
<registry>
  <item>
    <key>HKEY_CURRENT_USER\Software\Oracle\Test</key>
    <item name="Count" type="DWORD">400</item>
    <item name="Test" type="STRING">ABCDE</item>
  </item>
</registry>
<registry>
```

The SMC adds the Windows Registry key named Test in the directory named HKEY_CURRENT_USER\Software\Oracle branch and creates a String value named Test and a DWORD value named Count inside the key. If the same script is used in UNINSTALL mode, the SMC removes the key from the Registry.

ODBC

This section creates the ODBC driver and DSNs in the client device. For example,

```
<odbc>
  <item name="driver:Oracle Lite 40 ODBC Driver" dll='$APP_
DIR$\bin\olod2040.dll'>
    <version>02.00</version>
    <admin>$APP_DIR$\bin\olad2040.dll</admin>
  </item>
  <item name="driver:Oracle Lite 40 ODBC Driver (Client)" dll='$APP_
DIR$\bin\olcl2040.dll'>
    <version>02.00</version>
    <admin>$APP_DIR$\bin\olclad2040.dll</admin>
  </item>
  <item name="dsn:POLITE" driver='Oracle Lite 40 ODBC Driver' dll='$APP_
DIR$\bin\olod2040.dll'>
    <DataDirectory>$APP_DIR$\OLDB40</DataDirectory>
  </item>
</odbc>
```

JAVA

This section lists the JRE file name and the expected JAVA version. If the expected JAVA version is greater than the version that is already existing in the computer, the SMC installs a new JRE.

```
<java version="1.3.1">
  <item>
    <jre>webtogo\j2re-1_3_1_01-win.exe</jre>
    <iss>webtogo\jre_setup.iss</iss>
  </item>
</java>
```

LINK

This section creates Windows program links (or Program menu items). Each entry must have name, a program file name and a folder name. For example,

```
<link>
  <item name=' Oracle Web-to-go'>
    <folder>Startup</folder>
    <file>$APP_DIR$\webtogo\webtogo.exe</file>
  </item>
</link>
```

INI

This section creates entries in the INI (configuration) files. Each item must have an INI file name and a set of values to be added to a section. For example,

```
<ini>
  <item name="POLITE.INI" section="All Databases">
    <item name="DatabaseID">200</item>
    <item name="NLS_LANGUAGE">ENGLISH</item>
  </item>
</ini>
```

EXECUTE

This section lists all the programs to be executed during the installation process. Each item must have a program name, wait period, and program arguments. The wait is specified in milliseconds. For example,

```
<execute>
  <item>
    <file>$APP_DIR$\webtogo\webtogo.exe</file>
    <args>-h</args>
    <wait>WebToGoSetupExit/WebToGoSetupStop</wait>
  </item>
</execute>
```

REGISTER

This section lists all DLLs (or COM objects) to be registered with the Windows Operating System. For example,

```
<register >
  <item>$APP_DIR$\webtogo\msync_com.dll</item>
</register>
```

10.4 Device Management Server

This section presents a discussion of the Device Management Server and describes how to manage devices and their corresponding platforms using the Mobile Manager. Topics include:

- [Section 10.4.1, "Overview"](#)
- [Section 10.4.2, "Mobile Manager"](#)

10.4.1 Overview

The Device Management Server runs on the Oracle Database Lite Mobile Server. It contains a set of Java Servlets and a web based Device Management Application, which is a component of the Mobile Manager. To provide DM functionalities, DMS servlets interact directly with Device Management Clients. Some of the common functionalities of DMS are listed below.

- Oracle Database Lite Platform Installation
To install the Oracle Database Lite Client on a device, download the Mobile Client Setup program from the Mobile Server. The Setup program interacts with the DMS to install the Oracle Database Lite Client Platform on the device.
- Software Installation and Upgrades
To install, de-install, and upgrade applications, this feature works in conjunction with the Software Management Client.
- Send device commands.
- Enable or disable devices.
- Enable or disable Oracle Database Lite platforms.
- Create custom Oracle Database Lite platforms.

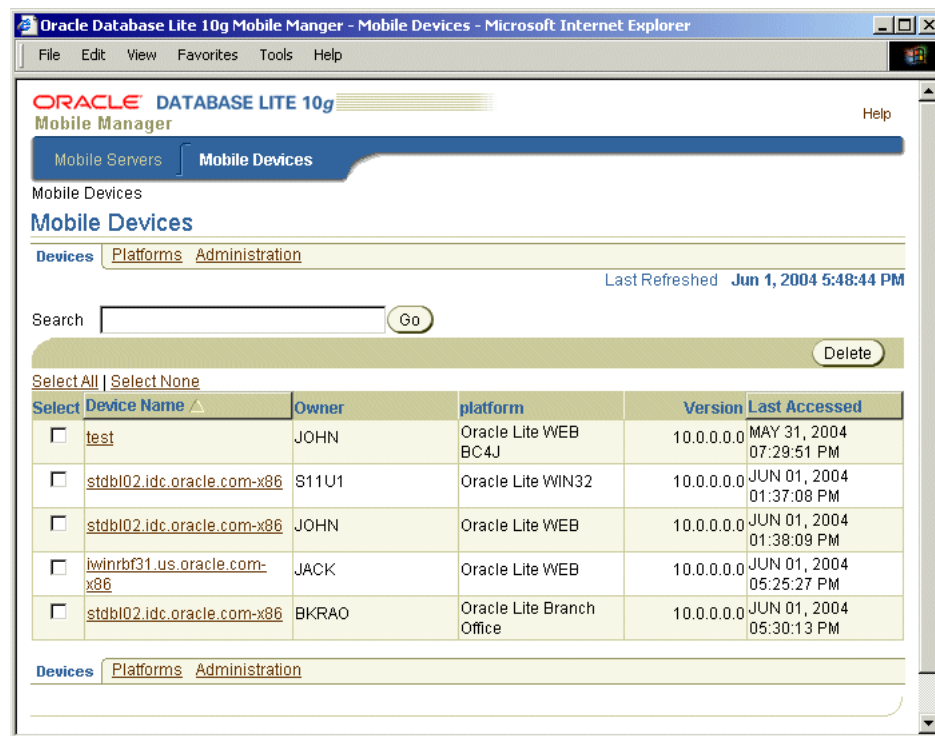
10.4.2 Mobile Manager

The Mobile Server's Mobile Manager application presents device management functionality. The following sections enable you to manage device platforms, commands, device configuration, and network providers. Topics include:

- [Section 10.4.3, "Managing Device Platforms"](#)
- [Section 10.4.4, "Managing Device Commands"](#)
- [Section 10.4.5, "Managing Device Configuration"](#)
- [Section 10.4.6, "Managing Network Providers"](#)
- [Section 10.4.7, "Administering Devices"](#)

10.4.3 Managing Device Platforms

To manage device platforms, login to the Mobile Server and click the Mobile Manager link in the workspace. The Mobile Server Farms page appears. Click the Mobile Devices in the top section. As [Figure 10-3](#) displays, the Devices page appears.

Figure 10–3 Devices Page

Using the Devices page, you can manage device platforms, commands, configuration, and network providers.

10.4.3.1 Modifying Platform Properties

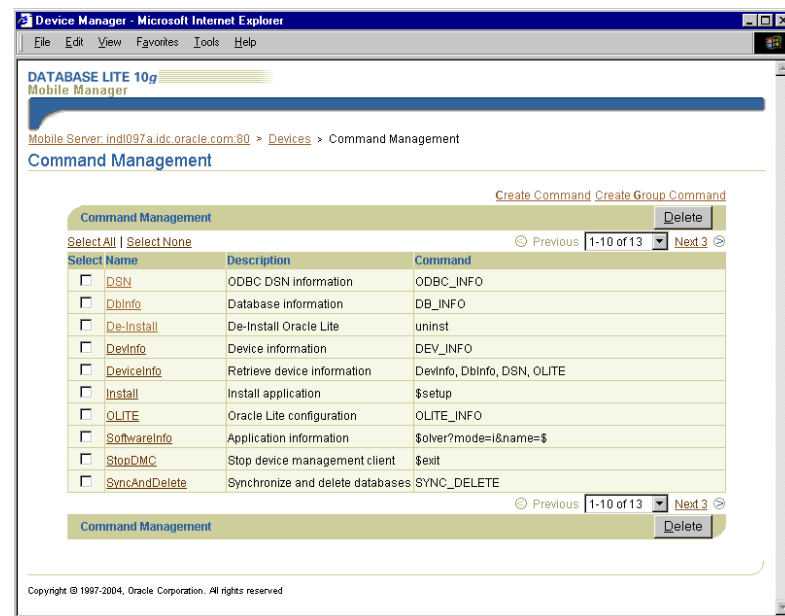
To modify device platform properties, select the Platform Name and click the Properties button. The View/Edit Properties page appears. Enter the path and file name of the modified Setup INF file and choose the required Bootstrap command from the list displayed. The Device platform is enabled by default. Click the Apply button. The Mobile Manager displays a confirmation message.

10.4.3.2 Creating Custom Platform

The Mobile Manager enables users to create custom device platforms as an extension to the existing platform. To create a custom platform, select the Platform Name and click the Extend button. The Extend Platform page appears. Enter the custom platform name, path and file name of the Setup INF file, and choose the required Bootstrap command. The Device Platform is enabled by default. Click the Extend button. The Mobile Manager displays a confirmation message.

10.4.4 Managing Device Commands

To manage device commands, navigate to the Administration page and click the Command Management link. As [Figure 10–4](#) displays, the Command Management page appears.

Figure 10–4 Command Management Page

Using the Command Management page, you can modify existing device commands and create new device commands.

10.4.4.1 Modifying Existing Commands

To modify existing commands, click the required Command Name link. The chosen Command's Properties page appears. Enter the command name, description, and syntax in the corresponding fields. To check the accuracy of the command syntax, click the Syntax Check button. If no errors are found, the Mobile Manager displays a confirmation message. Click the Apply button. The Mobile Manager displays a confirmation message.

10.4.4.2 Creating New Commands

To create new commands, click the Create Command button. The Create Command page appears. Enter a unique Command ID, Command String, and Description in the corresponding fields. Click the Create button. The Mobile Manager displays a confirmation message.

10.4.4.3 Creating Group Commands

To create group commands, click the Create Group Command button. The Create Group Command page appears. Enter a unique Command ID and description in the corresponding fields and select the required set of commands that you need to group. The Command Weight feature controls the order in which commands must be executed. For example, a command with Weight 1 is executed first and a command with Weight 2 is executed next. Users must specify a weight for all the commands for the chosen group command. Click the Add button. The Mobile Manager displays a confirmation message.

10.4.5 Managing Device Configuration

Client configuration parameters are used by the software installer to modify client side configuration files such as the `devmgr.txt`, `polite.ini`, and the `webtogo.ora`.

To manage Device Configuration parameters, click the Configuration Management link under the Administration section. As [Figure 10–5](#) displays, the Configuration Management page appears.

Figure 10–5 Configuration Management Page

Oracle Database Lite 10g Mobile Manager - Configuration Management - Microsoft Internet Explorer

File Edit View Favorites Tools Help

ORACLE DATABASE LITE 10g Mobile Manager

Mobile Servers Mobile Devices

Mobile Devices > Configuration Management

Configuration Management

File Name: dmc.inf Show Add Apply Delete

Select All | Select None

Select Name	Value
<input type="checkbox"/> PUSH_PORT	8082
<input type="checkbox"/> DISABLE_PROMPT	0
<input type="checkbox"/> HTTP_PROXY	\$HTTP_PROXY\$
<input type="checkbox"/> SERVER_URL	\$SERVER_URL\$
<input type="checkbox"/> USER_NAME	\$USER_NAME\$

Copyright © 1997-2004, Oracle. All rights reserved. Help

Using the Configuration Management page, you can modify existing configuration parameters or add new parameters.

10.4.5.1 Modifying Client Configuration Items

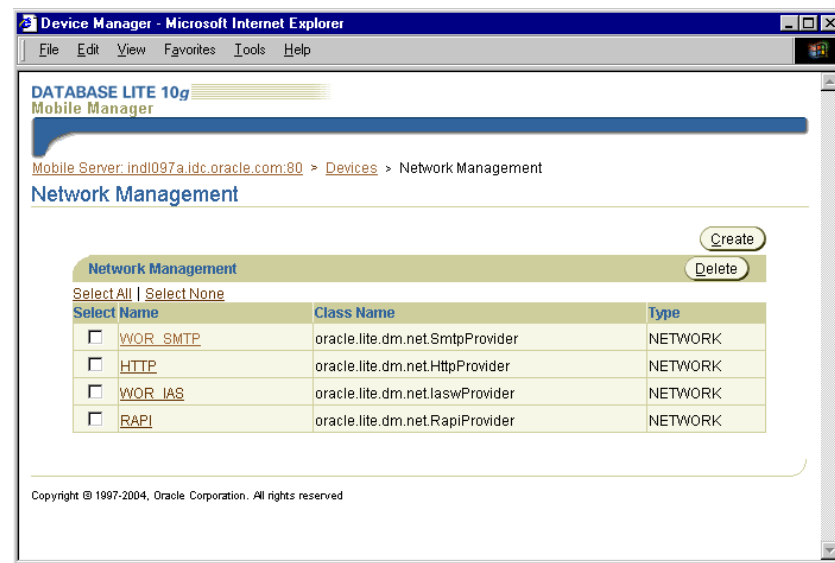
To modify client configuration items, enter the modified configuration value against the required configuration parameter and click the **OK** button. The Mobile Manager displays a confirmation message.

10.4.5.2 Creating Configuration Items

To create configuration parameters, select the configuration object name and click the Add button. Enter a unique configuration parameter name and value in the corresponding field and click the **Create** button. The Mobile Manager displays a confirmation message.

10.4.6 Managing Network Providers

The Network Management page enables you to modify properties of a network provider and register new network providers. To manage network providers, navigate to the Devices page, and click the Network Management link under the Administration section. As [Figure 10–6](#) displays, the Network Management page lists existing client configuration items.

Figure 10–6 Network Management Page

10.4.6.1 Modifying Network Provider Properties

To modify existing network providers, click the Network Provider name. The chosen Network Provider's properties page appears. Enter the modified Class Name and Meta Data in the corresponding fields. Click the Apply button. The Mobile Manager displays a confirmation message.

10.4.6.2 Creating New Network Providers

To register new network providers, click the Create button. The Create Provider page appears. Enter a unique Network Provider ID, class name, and meta data in the corresponding fields, and click the Create button. The Mobile Manager displays a confirmation message.

10.4.7 Administering Devices

This section enables you to modify device properties, create new device addresses, view device information and device application information. Topics include:

- [Section 10.4.7.1, "Modifying Device Properties"](#)
- [Section 10.4.7.2, "Viewing Applications"](#)
- [Section 10.4.7.3, "Viewing Device Information"](#)
- [Section 10.4.7.4, "Scheduling or Sending Commands"](#)
- [Section 10.4.7.5, "Viewing Device Logs"](#)
- [Section 10.4.7.6, "Viewing the Device Command History"](#)

10.4.7.1 Modifying Device Properties

Device Properties are comprised of the device address, network provider key, network provider protocol, and options to enable and upgrade a device. Using the Device Properties page, the Administrator can modify properties of a specific device and create a new device address. The Device Properties page provides controls to enable or

disable a device and automatically upgrade the device. After disabling the upgrade feature, the device does not receive any software updates.

To modify device properties, navigate to the Devices page and click the Platform Name link. The Mobile Manager displays a list of associated devices for the chosen platform. Click the required Device Name link. The Properties page appears. Enter the modified device address, network provider key, and network provider protocol in the corresponding fields. Click the Apply button. The Mobile Manager displays a confirmation message.

10.4.7.2 Viewing Applications

To view applications on a device, click the Applications link. The Applications page lists applications, version, setup time, and location details.

10.4.7.3 Viewing Device Information

The Mobile Manager displays general and database information for a chosen device. To view device information, click the Information link. The Information page contains general information about the Operating System, Oracle Database Lite platforms, networking, CPU, and storage.

To view database information, click the Database link. The Database information page contains Oracle Database Lite information such as DSN and configuration details on the client device. Information about the `ODBC.ini` and `POLITE.ini` is displayed under the Oracle Lite DSN section.

10.4.7.4 Scheduling or Sending Commands

The Commands page enables the Administrator to schedule or send device commands to a specific device. To schedule or send commands, click the Commands link. The Commands page appears.

Scheduling Commands

To schedule commands, click the Schedule button. The Schedule Command page appears. Using the Parameter section of the Command Schedule page, the Administrator can choose the required command, send the command priority, and when required, set extra command parameters. The Schedule section provides controls to set the command schedule start, repeat, and expiry durations. Enter the appropriate command schedule information in the corresponding fields, and click the Add button. The Mobile Manager displays a confirmation message.

Sending Commands

To send commands, choose the required command and click the Send Now button. To send the chosen command, the Mobile Manager seeks your confirmation and displays a confirmation message subsequently. The Mobile Manager displays an argument collection page when a command requires arguments. For example, the Upload File command requires a file name as an argument. To send the command to the device, click the Yes button.

10.4.7.5 Viewing Device Logs

The Mobile Manager displays device logs and synchronization logs. To view device logs, click the Device Logs link. The Device Logs page lists existing device logs. To view synchronization logs, click the Synchronization link.

10.4.7.6 Viewing the Device Command History

To view the Device Command History, click the Command History link. The Command History page lists a history of commands that were implemented for the chosen device.

This document enables the Mobile Server Administrator to set tracing parameters. Topics include:

- [Section 11.1, "Overview"](#)
- [Section 11.2, "Setting Trace Parameters"](#)
- [Section 11.3, "Editing the webtogo.ora File"](#)
- [Section 11.4, "Running the Mobile Server in Debug Mode"](#)

11.1 Overview

The Mobile Server enables the Administrator to turn the trace feature on and off as required and change the server's trace configuration, while the server is running. Based on the required trace output type, the Mobile Server administrator can choose a trace output destination.

When the Mobile Server is running in standalone mode, the Console option generates trace output to the Console window only, with the `-d0` command line option on. If you set this parameter to the option `-d0`, the trace output automatically appears on your console screen. Using the `-d0` option with this parameter overrides trace settings for other trace parameters, such as destination and level, in the `webtogo.ora` file. The `-d0` setting enforces the trace output to appear on your console screen instead of appearing in a file.

To generate trace output to a file, the Administrator can choose the File option. When required, the Administrator can also generate trace output to a remote machine. Trace output can then be viewed using the `wsh -m%` utility and the corresponding remote trace parameter. Applying changes that are associated with remote trace support automatically launch `wsh -m%TRACE_REMOTE_PORT%` in the console window. If the port number changes when a `wsh -m` process is running, the previous process is terminated immediately.

If you specify a valid value for the trace destination, the specified destination is used. If the necessary parameters for the specified destination are missing, then the default value for such parameters are applied to ensure that the trace output is generated to the default destination. If the default value for the specified parameter is missing, the default trace destination is automatically chosen, depending on the mode of the Mobile Server.

For more information on Mobile Server modes, see [Section 11.2.1, "Mobile Server Modes"](#).

For a description of the Tracing parameters and information on how to set them, see [Appendix B, "Mobile Server Configuration Parameters"](#).

11.2 Setting Trace Parameters

To set trace parameters, perform the following steps.

1. Login to the Mobile Server and click the **Mobile Manager** link in the workspace. The Mobile Server farms page appears. Click the following links. Your Mobile Server -> Administration -> Trace Setting. As [Figure 11–1](#) displays, the Trace Settings page appears.

Figure 11–1 Trace Settings Page

Using the Trace Settings page, you can choose to generate trace output, specify the trace output destination to the local console, file or remote machine. The Trace Settings page provides system and user filters to generate trace output to the required system level and choose users that require the trace output generation utility.

2. As [Table 11–1](#) describes, enter data in the Trace Settings page.

Table 11–1 Trace Settings Page Description

Field	Description
Trace Output	To generate trace output, click this box and choose Yes.
Console	To specify the local console as the trace output destination, select this option.
File	<p>To specify the trace output file (error.log) as the trace output destination, select this option.</p> <p>If selected, enter the Trace Base File Name, Trace File Size (in MB), and Trace File Pool Size in the corresponding fields. To create a trace file for every user, click the Create Trace File for Every User box and choose Yes. The Trace Settings page displays default values for these fields.</p>

Table 11–1 (Cont.) Trace Settings Page Description

Field	Description
Remote	To specify trace output to a remote destination, select this option. If selected, enter the Trace Monitor Host name and Trace Monitor Port number in the corresponding fields. For more information on Mobile Server modes, available trace destinations, and default trace destinations, refer Section 11.2.1, "Mobile Server Modes" .
HTTP Request	To generate HTTP output and webtogo trace information as trace output, select this option.
SQL Statements	To generate Consolidator's SQL queries as trace output, select this option.
Java Methods	To generate all <code>system.out</code> output from the Mobile Server and Consolidator's Java methods, select this option. Note: The Mobile Server automatically filters exceptions and errors as trace output to a Mandatory level. Mandatory trace output is automatically generated, only when the trace feature is turned on.
All Users	To generate trace output for all users, select this option.
No Users	If selected, does not generate trace output for users.
Selected Users	To generate trace output for specified user names only, select this option. In the Selective Users field, enter comma separated user names. For example, arnold, john, jack.

3. Click the **Apply** button.

11.2.1 Mobile Server Modes

[Table 11–2](#) lists all Mobile Server modes, the available trace destinations, and the corresponding default trace destinations.

Table 11–2 Trace Destinations for Mobile Server Modes

Mobile Server Mode	-d0 Option	Available Trace Destinations	Default Trace Destination
Standalone	Overrides webtogo.ora setting	Console, File, Remote	Console
NT Service	N/A	File, Remote	File
Oracle9iAS	N/A	File, Remote	File

11.3 Editing the webtogo.ora File

To edit the webtogo.ora file, navigate to the Administration page and click the **Edit Config File** link. As [Figure 11–2](#) displays, the Edit Configuration File webtogo.ora page appears.

Figure 11–2 Edit Configuration File webtogo.ora Page

Edit the webtogo.ora file in the required areas and click **Apply**.

Using the webtogo.ora File or Command Line -d Option

To specify the required trace output level, the administrator can optionally use the Mobile Server or the webtogo.ora file or the Command Line -d option. The administrator must specify a number that represents all the levels filtered in, as the number is a Bitwise OR operation of the chosen levels. To display trace output for all three levels, you must use 1 | 2 | 4 = 7. To be compatible backward, you must use the value 0 instead of the value 7, which hides all trace output.

If you set this parameter to the option -d0, the trace output appears on your console screen without appearing in a file, because using the -d0 option with this parameter overrides the trace settings for other trace parameters, such as destination and level, in the webtogo.ora file. The -d0 setting enforces the trace output to appear on your console screen instead of appearing in a file.

For more information on Mobile Server configuration parameters, see [Appendix B, "Mobile Server Configuration Parameters"](#).

11.4 Running the Mobile Server in Debug Mode

This document enables the Mobile Server Administrator to debug the Mobile Server and generate debugging output.

To run the Mobile Server in debug mode, perform the following steps.

1. Open the file **webtogo.ora** from the following location.


```
mobile\server\bin
```
2. Locate the following parameters and change their existing values as given below.
 - a. TRACE_ENABLE=Yes

- b. TRACE_DESTINATION=FILE
- c. TRACE_FILE_NAME=trace.log

Note: Debugging output is not generated when you set the Trace Destination value to Console.

The following log files can be used to debug problems while running applications in OC4J.

1. Use this log file to view OC4J server level output messages.
`<OC4J_HOME>\log\server.log`
2. Use this log file to view HTTP requests handled by the server.
`<OC4J_HOME>\log\http-web-access.log`
3. Use this log file to view exceptions or errors that are handled by OC4J.
`<OC4J_HOME>\application-deployments\webtogo\application.log`
4. Use this log file to view the file `trace_sys1.log` and other log files that are generated by the Mobile Server in the same directory.
`<OC4J>_HOME>\application-deployments\webtogo\trace_sys1.log`

Note: All output generated by the parameter `JupDebug` in the Mobile Server is recorded in the above listed log files.

This document enables the administrator to deploy Oracle Database Lite 10g. Topics include:

- [Section 12.1, "Overview"](#)
- [Section 12.2, "How To Deploy"](#)

12.1 Overview

Before using Oracle Database Lite 10g for the first time, end users must deploy the software on their client machines. The application deployment phase starts after the developer packages mobile applications and the Administrator publishes them into the Mobile Server Repository.

As the end user, you deploy the software only once for each Mobile Client that you use, by primarily performing the following tasks. Detailed steps are explained in subsequent sections of this document.

1. Install the appropriate Mobile Client on your client machine. For example, you can install the Mobile Client for Web-to-Go or the Mobile Client for Win32 on your Windows 32 client machine.
2. Synchronize the first time that you use Oracle Database Lite 10g. This first time synchronization process downloads the mobile applications (which your Administrator has subscribed you to as a user) and the data to your Mobile Client, enabling you to launch your applications from your Windows 32 machine or from your mobile device. It also initializes the synchronization system between your Mobile Client and the Mobile Server.

Note: Synchronization as such is a separate process, which also takes place outside of deployment. You synchronize the data periodically between the main (or central) Oracle database and your Oracle Lite database. For more information about synchronization, see the *Oracle Database Lite Developers Guide*.

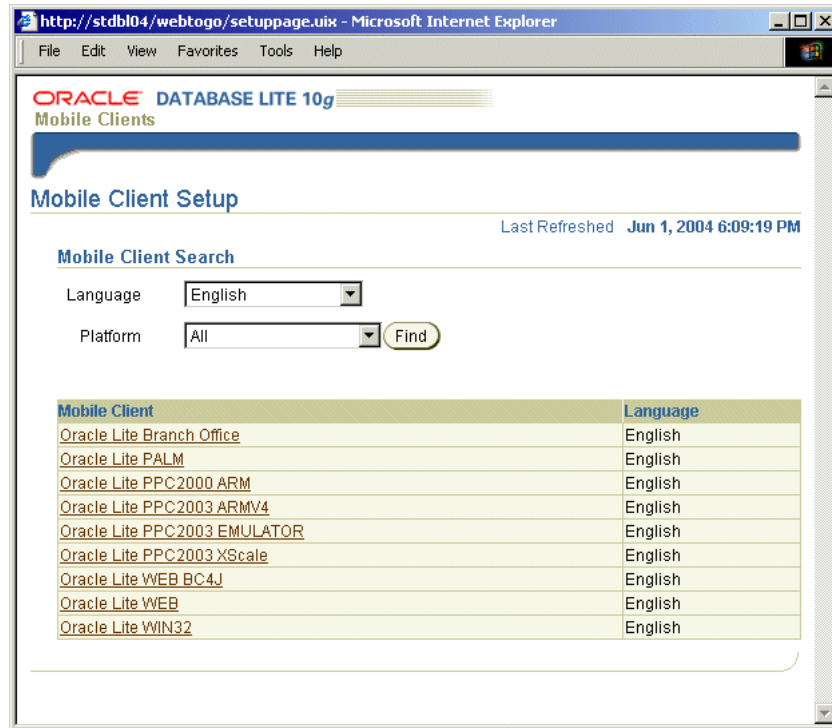
12.2 How To Deploy

To deploy Mobile Clients, perform the following tasks.

1. Open your browser and point it to the following URL.
`http://<mobile_server>/webtogo/setup`

As Figure 12–1 displays, the Mobile Client Setup page contains links to all the Mobile Clients.

Figure 12–1 Mobile Client Setup Page



2. Click the Mobile Client that you need for your chosen platform.
3. The Save As dialog box appears. The file name field displays the relevant executable setup file for the selected platform as a .cab, .exe, or .sis file type. Save the executable file to a directory of your choice.

Note: For the Palm device, download the Mobile Client for Palm to the directory on a Windows 32 machine that is used by the Palm desktop software to synchronize applications and data. Then do a hotsync to install the Mobile Client for Palm.

4. Install the Mobile Client. For platforms other than Palm, go to the corresponding directory where you saved the executable setup file. Double-click the file to execute it.

Note: For Web-based applications, the default installation is in the sync mode, not in the online/offline mode.

5. Synchronize the Mobile Client.

For the Mobile Client for Web-to-Go, this step (the synchronization step in deployment) takes place when you click **Next**, after executing the **setup.exe** and login to the Mobile Client for Web-to-Go.

For Mobile Clients of all other platforms, perform the following steps.

1. Locate the directories where you installed the runtime libraries, and launch the Mobile Sync application.
2. The **msync** dialog appears. Enter your user name and password. If you do not know your user name and password, ask your system administrator, who creates users and assigns passwords to each user. In the **Server** field, enter the URL for your Mobile Server. Click **Apply** and click **Sync**.

By completing the final step in deployment, Step 5, you have successfully deployed the Mobile Client.

For more information on how to install and use the Mobile Client for Web-to-Go, see [Chapter 14, "Deployment: Using the Mobile Client for Web-to-Go"](#).

Deployment: Creating an Install CD for Mobile Clients

This document describes how to create an install CD for installing Mobile Clients offline. Topics include:

- [Section 13.1, "How to Create an Offline CD for Mobile Clients"](#)
- [Section 13.2, "Installing a Mobile Client from the CD"](#)

13.1 How to Create an Offline CD for Mobile Clients

As the Administrator, you can create a set of files on a Windows machine. To enable users to install a Mobile Client offline, this file set can be burned to a CD. The files on the CD, install a Mobile Client and the requisite Java files.

This method is recommended for users who want to install a Mobile Client, but are connecting over low-bandwidth connections, although, they must still use such connections to synchronize.

The following sections describe how to download the offline install files.

13.1.1 Downloading the Offline Install Files

To download the offline install files, perform the following steps.

1. Using the following URL, connect to the Setup page.

`http://<mobile server>/webtogo/setup`

The Mobile Client Setup page lists Mobile Clients for all platforms.

2. Click the Mobile Client to download the Mobile Client **setup.exe** and save it in a local directory on your desktop.

13.1.2 Compile the Offline Install Files

To compile the files, perform the following steps.

1. Start the Command Prompt window and run the **setup.exe** from the directory that you specified in Step 2, using the **-distribution** flag as follows.

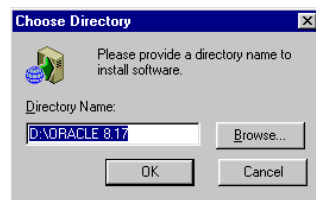
`C:\CDSETUP\setup.exe -d`

Note: This program creates a **setup.exe** icon in the folder that you selected in **Step 2**. Running this without the **-distribution** flag, by double-clicking the icon, for instance, performs a normal setup of the client on your machine. The **-distribution** option must be used to create the necessary files for an offline install.

2. You will be prompted for your User Name and Password. Enter your user name and password.
3. The **Choose Directory** prompts you for a destination directory where you want to create the CD-ROM image. Ensure that you create a new destination directory other than the directory in which the setup.exe file resides. The destination directory can be in the same path as the directory where the setup.exe file is, but it must be a different directory. Click **Browse** to locate the destination directory or enter the directory name, and click **OK**.

Figure 13–1 displays the Choose Directory dialog.

Figure 13–1 Choose Directory Dialog



4. All the requisite files are copied to the destination directory, pre-configured for an offline installation. These include an **.ini** file, which describes the Mobile Client and Java files which are required to upgrade the Java file versions on the client machine, including the **setup.exe** file.
5. Use this directory, or the files in this directory, to create a CD using a conventional CD burner, as CD writing software is not provided with Oracle Database Lite 10g.

13.2 Installing a Mobile Client from the CD

You can distribute the CD that you created as needed to users who want to install a Mobile Client. Users requiring a Mobile Client installation from this CD must run the **setup.exe** file on the CD. Once the Mobile Client is installed on the user's client machine, the user can connect to an instance of the Mobile Server normally.

Deployment: Using the Mobile Client for Web-to-Go

This document enables end users of mobile applications to install and use the Mobile Client for Web-to-Go. Topics include:

- [Section 14.1, "Overview"](#)
- [Section 14.2, "Installing the Mobile Client for Web-to-Go"](#)
- [Section 14.3, "Client-Side Configuration for Secure Socket Layer \(SSL\)"](#)
- [Section 14.4, "Support for Non-SSL Mobile Clients"](#)

14.1 Overview

After the mobile application developer packages a web based mobile application and the administrator publishes it, you, as the end user, are ready to use the mobile application. But, before you can use the application, you must first install the Mobile Client for Web-to-Go on your Windows 32 client machine. At this stage, you must familiarize yourself with the Web-to-Go Workspace.

14.2 Installing the Mobile Client for Web-to-Go

This section enables you to install the Mobile Client for Web-to-Go. Topics include:

- [Section 14.2.1, "Hardware Requirements"](#)
- [Section 14.2.2, "Software Requirements"](#)
- [Section 14.2.3, "Installation"](#)

You can install the Mobile Client for Web-to-Go by running the Web-to-Go setup program in your web browser. Before you install the Mobile Client for Web-to-Go, you must ensure that your environment contains the requisite hardware and software components.

14.2.1 Hardware Requirements

The Mobile Client for Web-to-Go requires the following hardware.

1. Pentium III
2. 12 MB Application RAM (Web-to-Go: 1.5 MB, Java: 10 MB)
3. 10 MB Hard Disk Space
4. 800 x 600 Display

14.2.2 Software Requirements

The Mobile Client for Web-to-Go requires the following software.

1. Microsoft Windows 95, 98, NT 4.0 (service pack 3 recommended), or 2000.
2. Netscape 4.0.6 or Microsoft Internet Explorer 4.0 with JavaScript enabled.

14.2.3 Installation

To install the Mobile Client for Web-to-Go, perform the following steps.

1. To connect to the Mobile Server, open a browser and enter the following URL.

`http://<mobile_server>:<port_number>/webtogo/setup`

Note: You must replace the <mobile_server_name> variable with the host name of your Mobile Server. If the Mobile Server is not accessible after specifying its name, you may need to enter the port number.

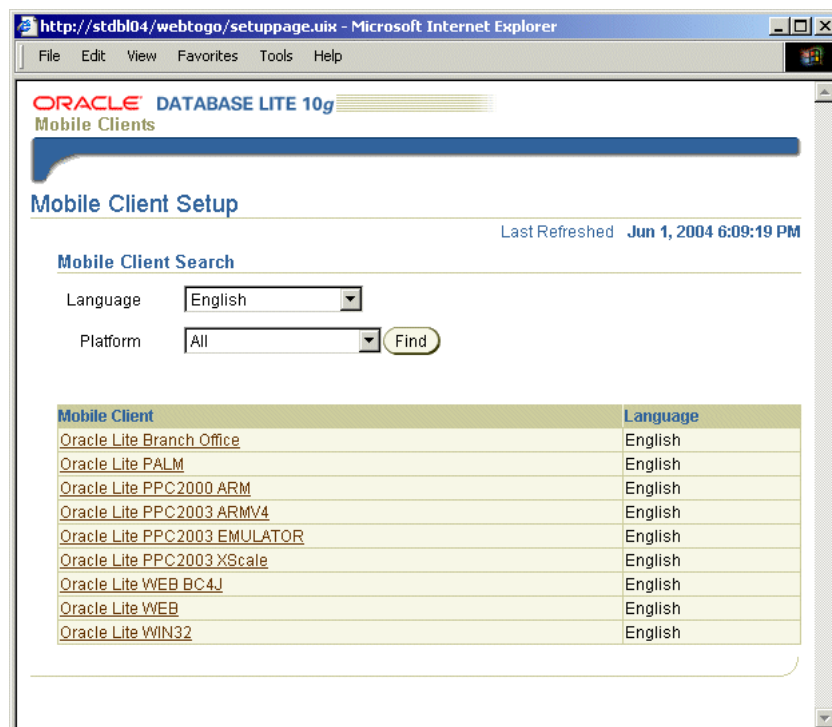
The default port number for the Mobile Server is 443. If your Mobile Server is running on port number 443, you do not have to specify the Mobile Server port number in the URL. If your Mobile Server is running on a port number other than 443, you must specify the Mobile Server port number in the URL as follows.

`https://<mobile_server_name>:<port_number>/setup`

The Setup page lists Mobile Client programs that are available for downloading.

Figure 14–1 displays the Mobile Client Setup page.

Figure 14–1 Mobile Client Setup Page



2. Click the required Mobile Client link from the list displayed.
3. If your browser is Netscape, choose a location to download the setup program, and click **OK**. In Windows Explorer, double-click **setup.exe** to run the setup program.

OR

If your browser is Internet Explorer, run the setup program from your browser window. Once started, the setup program prompts you to provide an install directory. Choose a directory, for example, `D:\Mobile Client`, and click **OK**.

The setup program downloads all the required components and starts the Mobile Client for Web-to-Go on your machine. Upon completion of the installation, the Web-to-Go screen appears.

14.3 Client-Side Configuration for Secure Socket Layer (SSL)

As the end user, you can configure the Mobile Client for Web-to-Go to establish an SSL connection between the Mobile Client and the Mobile Server. Oracle Database Lite 10g does not, however, support SSL connection between the browser and the Mobile Client for Web-to-Go. Topics include:

- [Section 14.3.1, "Communication between the Mobile Client and the Mobile Server"](#)
- [Section 14.3.2, "Connection between the Browser and the Mobile Client for Web-to-Go"](#)

14.3.1 Communication between the Mobile Client and the Mobile Server

Based on whether or not you download the Mobile Client for Web-to-Go from the Mobile Server running in SSL, you can choose to configure communication between the Mobile Client for Web-to-Go and the Mobile Server. The following sections provide a description of configuring communication between the Mobile Client and the Mobile Server.

Mobile Client Download from a Mobile Server which is Running in SSL Mode

The Mobile Client for Web-to-Go which is downloaded from the following URL is automatically configured for SSL and does not require manual configuration on the part of the end user. This download enables the Mobile Client to communicate with the Mobile Server in SSL mode.

```
https://<mobile_server>:<port>/setup
```

Mobile Client Download from a Mobile Server which is not Running in SSL Mode

If you have downloaded the Mobile Client for Web-to-Go from a Mobile Server, which, is not running in SSL mode, you must modify the `SERVER_URL` parameter in the configuration file **webtogo.ora** as follows.

```
SERVER_URL=https://<mobile_server>:<port>/webtogo/setup
```

Note: in the location bar, you must type `https`, to specify and indicate the SSL Mode, and not `http`.

14.3.2 Connection between the Browser and the Mobile Client for Web-to-Go

While trying to connect to the Mobile Client for Web-to-Go in SSL mode, you will not be able to connect to the Mobile Client, even if the following conditions exist.

1. The Mobile Server is running in SSL mode, as a module of Oracle9iAS.
2. The Mobile Client for Web-to-Go is also running in SSL mode.

To connect to the Mobile Client for Web-to-Go using a browser, you must specify HTTP and not HTTPS in the client URL, although the communication between the client and the server is through the HTTPS protocol.

For example, `http://<client_machine>/webtogo`

14.4 Support for Non-SSL Mobile Clients

Mobile Servers running in SSL mode possess the ability to synchronize with any Mobile Client which is running in SSL or non-SSL mode. But, in the case of the Mobile Client for Web-to-Go, the client also runs in SSL mode to synchronize with the Mobile Server, which is running in SSL mode.

As SSL is not supported on many Mobile Clients, the Mobile Server, which, is running in SSL mode, still supports Mobile Clients that are running in non-SSL mode.

Note: Inside Oracle9iAS, the Mobile Server runs on both SSL and non-SSL ports, to support SSL and non-SSL clients. The Oracle9iAS must be configured to run on both SSL and non-SSL ports, as a default function.

Administering and Using the Branch Office

This document describes how to install, configure, administer and use the Mobile Client for Branch Offices. Topics include:

- [Section 15.1, "Introduction"](#)
- [Section 15.2, "Branch Office Installation and Configuration"](#)
- [Section 15.3, "Architecture"](#)
- [Section 15.4, "Administration"](#)

15.1 Introduction

This document introduces the Oracle Database Lite Branch Office. Topics include:

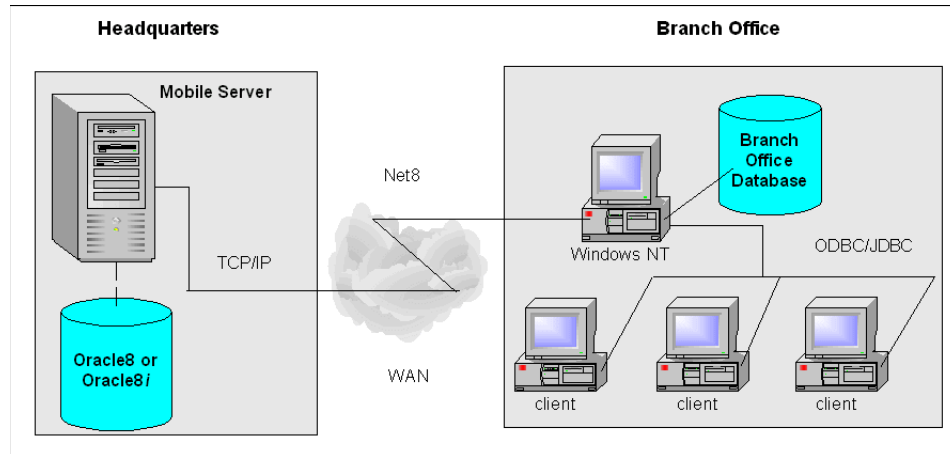
- [Section 15.1.1, "What is the Branch Office?"](#)
- [Section 15.1.2, "How the Branch Office Works"](#)
- [Section 15.1.3, "The Branch Office Manager"](#)
- [Section 15.1.4, "Synchronizing Data with Headquarters"](#)

15.1.1 What is the Branch Office?

The Branch Office provides access to the Branch Office database for 16 concurrent networked users. It enables the deployment of enterprise data and applications to geographically distributed sites that are running a Branch Office database. Each Branch Office database is centrally managed and supports multiple client connections, thereby eliminating local database administration tasks.

The Branch Office database synchronizes client data with the Oracle database at the company headquarters. [Figure 15–1](#) illustrates the Branch Office database at a Branch Office location and its connection to the Oracle database server at the headquarters. Branch Office clients connect to the Branch Office database using either ODBC or JDBC connections. Clients access and update the Branch Office database which contains a subset of the corporate database located at the company's headquarters.

Figure 15–1 Branch Office Overview



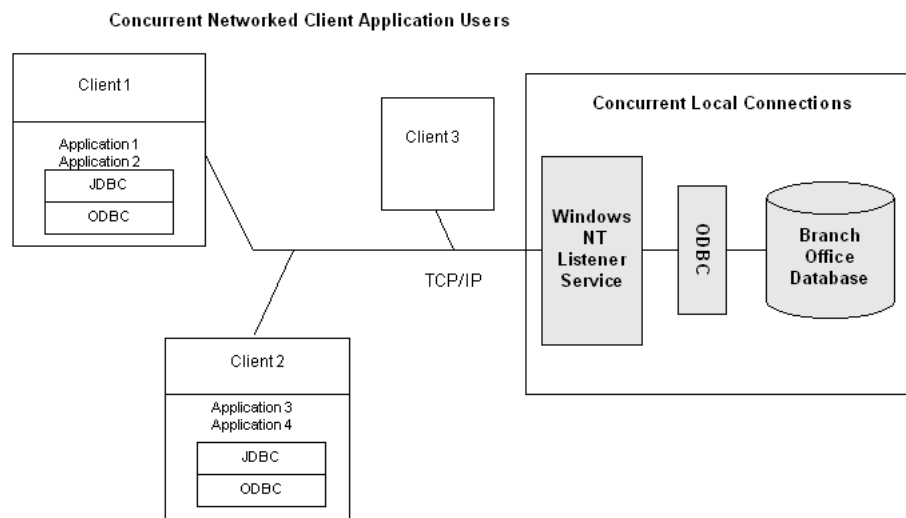
15.1.2 How the Branch Office Works

Each Branch Office database supports 16 concurrent networked users (Branch Office clients). These clients do not require a connection to their company headquarters and are allowed to work independently, without the corporate database.

The Branch Office also supports 16 concurrent local ODBC/JDBC connections to the Branch Office database. These local connections can be used for applications which perform such background tasks as reporting, mass changes or updates, and bulk data loading.

As Figure 15–2 displays, Branch Office clients and local ODBC and JDBC applications access the Branch Office database simultaneously. Multiple applications can run on each client.

Figure 15–2 Accessing the Branch Office Database



15.1.3 The Branch Office Manager

The Branch Office requires no local database administration and enables configuration and monitoring of Branch Office database services and users. Using a web based interface called the Branch Office Mobile Manager, the Branch Office Administrator can centrally manage Branch Office operations. Furthermore, the Branch Office Mobile Manager lowers administration costs by eliminating the need for a System Administrator to be physically present at each Branch Office.

15.1.4 Synchronizing Data with Headquarters

Data synchronization for Branch Offices is centrally managed by the Branch Office Administrator. The Administrator can synchronize applications and data with the database located in the headquarters, through a TCP/IP connection. Synchronization between the Branch Office database and the headquarters database is executed through the Mobile Server. For more information on synchronizing data, see the *Oracle Database Lite Developer's Guide*.

The centralized management and data synchronization between the headquarters and its branches enables each Branch Office to synchronize data with the corporate database according to a pre-determined schedule. This allows for data replication based on geographic factors and alternate time zones.

Data specific to a given Branch Office is synchronized from the corporate database server to the Branch Office database. Each Branch Office database represents a single instance of replicated data and is the physical data repository that is accessed by Branch Office clients.

15.2 Branch Office Installation and Configuration

This section describes how to install and configure the Mobile Client for Branch Offices. Topics include:

- [Section 15.2.1, "Terms and Concepts"](#)
- [Section 15.2.2, "Overview"](#)
- [Section 15.2.3, "Branch Office Installation"](#)

15.2.1 Terms and Concepts

Branch Office

A deployment concept of Oracle Database Lite designed for remote offices and Branch Office configuration.

Mobile Client for Branch Office

Self contained bundle of Oracle Database Lite libraries installed in the Branch Office machine.

Branch Office Administrator

Logical user responsible for the management of Branch Office users, data and applications.

Branch Office Database

Multi user of the Oracle Lite database.

Branch Office Application

Native or Java applications that access the Branch Office database over remote ODBC or JDBC connections.

Branch Office User

Logical user who is a client of the Branch Office database.

Branch Office Administrators

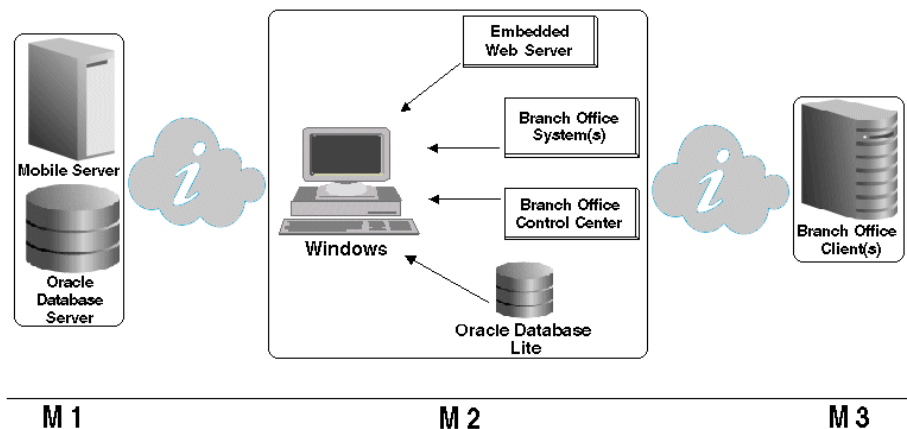
Group of Branch Office Administrators who are managed by the Mobile Manager.

15.2.2 Overview

To help understand and successfully implement a Branch Office setup, this section presents a sample setup that simulates a typical Branch Office environment. As [Figure 15–3](#) displays, this process assumes that the Branch Office configuration has the following installations on three machines.

1. **M 1: One Mobile Server** with a corporate Oracle database server.
2. **M 2: One or many Branch Office system(s)** running on a Windows machine. This setup includes an embedded web server, a multi user Oracle Lite database, and the Branch Office Mobile Manager. The Branch Office libraries are installed as part of the Mobile Client for Branch Office.
3. **M 3: One or a maximum 32 Branch Office Clients** that host the Branch Office application (.exe) and uses a remote ODBC/JDBC connection to access data in the Branch Office (multi user) database located on M 2.

Figure 15–3 Branch Office Setup



The following section enables you to install and configure the Branch Office.

15.2.3 Branch Office Installation

To install and configure the Branch Office, perform the following steps.

1. Install the Mobile Server on the machine named M1.
2. Using the Packaging Wizard, package the Branch Office application. During the application packaging process, select *Oracle Lite Branch Office* as your target

platform. For more information on how to package your applications using the Packaging Wizard, refer the *Tools and Utilities Guide*.

3. Using the Mobile Manager's **Applications** page, publish your Branch Office application. Select the Branch Office application that you need to publish and click the **Publish Application** button.
4. Using the Mobile Manager's **Users** page, create a Branch Office Administrator user and add this user to the *Branch Administrators* group. You must provide administrator privileges to the Branch Office Administrator user.
5. Using the Mobile Manager's **Applications** page, click the published Branch Office application link. Click the **Files** tab and choose the application files that you want installed on the Branch Office Client (M3) machine. Using the **Make Public** button, provision these files as public files.
6. Provide access privileges to the Branch Office *Administrator* user. Using the Mobile Manager's **Applications** page, select the Branch Office application that you need to provide user access privileges for and click the **Access** link. Grant access privileges by selecting the check box displayed against the Branch Office Administrator.
7. Using the following URL, download and install the Mobile Client for Branch Office onto the Branch Office machine (M2).

`http://<mobileserver>/webtogo/setup`

8. Using the following URL, open a browser window in the Branch Office machine (M2) and connect to the local Branch Office Web Server using the appropriate Branch Office Administrator user name and password.

`http://<branch_office_hostname>`

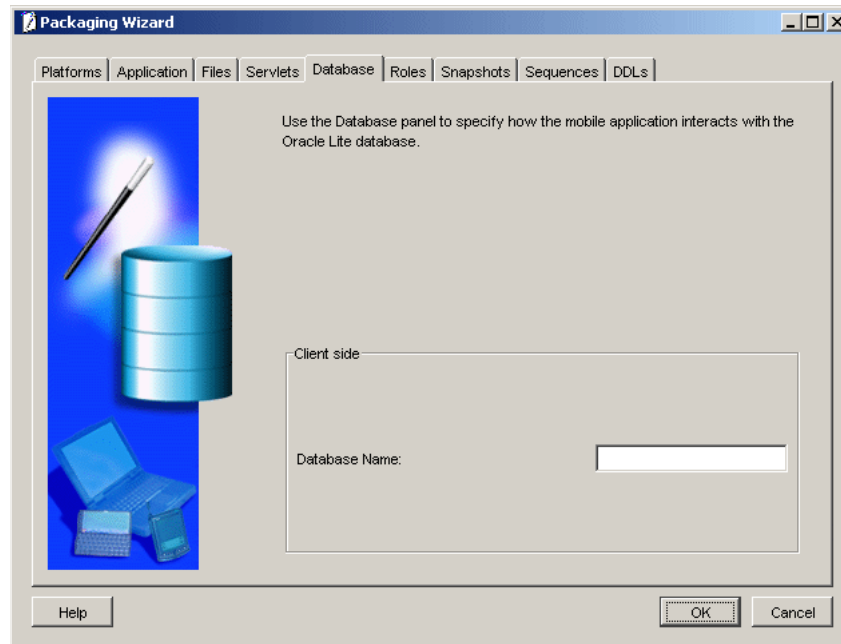
OR

`http://localhost/webtogo`

Note: Normally the Branch Office Web Server is automatically started by the setup program. If not, open the Control Panel and choose Services. Start the service name *Oracle Web-to-Go*.

9. At this stage, the Branch Office performs a complete synchronization process with the Mobile Server.
 - a. A directory is created under the directory **o1db40** with the name of your Branch Office Administrator user name. Under this directory, the Branch Office creates a database file with the same name as the **Database Name** provided in the Packaging Wizard as displayed in [Figure 15-4](#).

Figure 15–4 Database Panel - Packaging Wizard



- b. The Branch Office automatically creates a DSN entry in a <user name>_<database name> format.

For example, if your Branch Office Administrator user name is Tom and your database name is BranchDB, the Branch Office creates a DSN entry named Tom_BranchDB.

This stage marks the conclusion of the Branch Office Installation and Configuration process. In the next step, you must configure the Branch Office Client (M 3).

10. On the Branch Office Client machine (M3), open a browser window using the following URL and download the *ODBC Driver* program. This action creates a DSN entry in the Branch Office Client machine (M3) with the same name as the **Database Name** provided in the Packaging Wizard.

`http://<branch_office_hostname>/public/download`

Note: The DSN name on the Branch Office Client machine (M3) created by the ODBC Driver program is different from the DSN name on the Branch Office machine (M2).

11. On the Branch Office Client machine (M3), open a browser and download the Branch Office application files using the following URL.

`http://<branch_office_hostname>/public/download`

12. On the Branch Office machine M3, add the following DLL's to your path.

- oladc12040.dll
- olc12040.dll

13. Open the Branch Office Mobile Manager and create a new Branch Office user. The remote JDBC connection string for the user named "boUser" is given below.

```
<boUser>/<boUser password>@jdbc:odbc:<DSN>
```

Note: The DSN name is the same as the **Database Name** provided in the Packaging Wizard. It can be located in the ODBC .INI file on the Branch Office Client machine (M3). The DSN points to the remote database listener (port 100) located on the Branch Office machine (M2).

15.3 Architecture

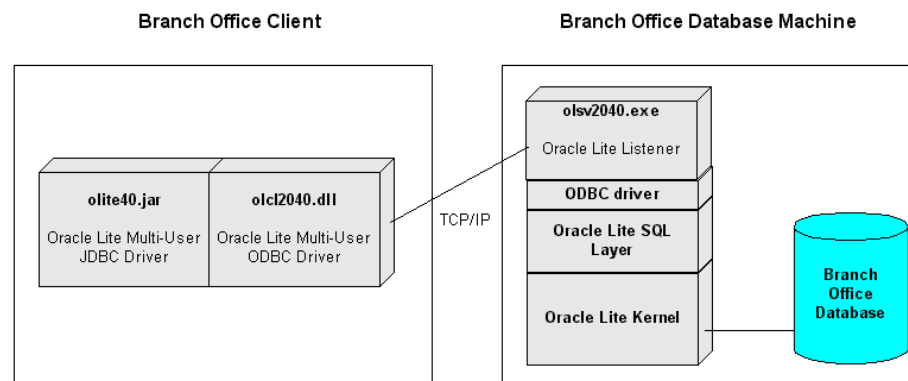
This chapter describes the components of the multi user architecture. Topics include:

- [Section 15.3.1, "The Branch Office Environment"](#)
- [Section 15.3.2, "Connecting Clients to the Branch Office Database Machine"](#)

15.3.1 The Branch Office Environment

The Branch Office environment is comprised of two parts. As [Figure 15–5](#) displays, they are the Branch Office Client component and the Branch Office Database component.

Figure 15–5 The Branch Office Environment



15.3.1.1 The Branch Office Client

The Branch Office client machine executes both ODBC and Java based applications which access the Branch Office database (**branch.odb**). The Branch Office includes the following components.

- [ODBC Driver](#)
- [JDBC Driver](#)

ODBC Driver

The client ODBC driver (**olcl2040.dll**), supports ODBC based client application connections for the Branch Office database. This driver connects ODBC applications to the Branch Office database. Based on the parameters specified in the client DSN, it searches for the Windows NT service running on the Branch Office database machine.

JDBC Driver

Java based client applications connect to the Branch Office database through a JDBC connection. This JDBC driver (**olite40.jar**) uses the ODBC driver for the connection. The ODBC driver makes the actual connection for the JDBC client application by first reading the DSN defined parameters and then by searching for the associated Windows NT service.

15.3.1.2 The Branch Office

The Branch Office contains the following components.

- [Branch Office Database](#)
- [Oracle Database Lite Listener Service](#)
- [Mobile Client for Web-to-Go](#)

Branch Office Database

The Branch Office database machine is the interface between Branch Office clients and the database at company headquarters.

The Branch Office database (**branch.odb**) is a file created by the Mobile Server during synchronization. This database file is a subset of the headquarters database. Its tables are built on the headquarters database server. The Branch Office database file does not support Oracle Database Lite utilities such as `CREATEDB` or `REMOVEDB`. In addition, the Branch Office database cannot be created using the Oracle8 Navigator.

The Mobile Server Packaging Wizard defines and generates replication support for tables. For more information, see the *Oracle Database Lite Developer's Guide*.

Note: Snapshots are owned by `SYSTEM`. The password is the Branch Office administrator's password.

Oracle Database Lite Listener Service

The Branch Office database machine contains the ODBC listener service named **olsv2040.exe**. This process creates a separate connection to the Branch Office database for every client network connection.

The listener service is dependent on Java. Before starting the listener service, the database machine must have the JavaSoft Java Runtime Environment (JRE) Version 1.2.2. The JRE can be downloaded from the Java technology web site.

The system `PATH` variable must include a path reference to this `bin` directory.

Mobile Client for Web-to-Go

The Mobile Client for Web-to-Go runs on the Branch Office database machine and acts as a web server to run the Branch Office Mobile Manager. This feature allows the system administrator to access the Branch Office Mobile Manager and maintain a Branch Office database without being physically present at the Branch Office.

The Mobile Client for Web-to-Go enables users to deploy applications on client machines, using a browser which points to the Branch Office database machine. The Mobile Client for Web-to-Go publishes client applications as public files, so that client applications that use the Branch Office database can be downloaded directly from the Branch Office database machine. The Mobile Client for Web-to-Go executes as a background process to support browser based applications and distribution of public files.

15.3.1.3 Company Headquarters

The Oracle Database server resides at the company headquarters. The Mobile Client for Web-to-Go on the Branch Office database machine synchronizes all data changes in the Branch Office database with the database located at the headquarters.

15.3.2 Connecting Clients to the Branch Office Database Machine

The client applications connect with the Branch Office database machine via TCP/IP. The client driver, **olcl2040.dll**, facilitates this communication by connecting with the **olsv2040.exe** listener service on the Branch Office database machine. For every client connection, the listener service establishes a separate connection thread with the Branch Office database, **branch.odbc**.

Establishing concurrent client connections requires that the listener service on the Branch Office database machine be started before the network connections are established.

15.3.2.1 ODBC Connection

To make a client connection to a Branch Office database, you must first set up an ODBC data source name (DSN) using the ODBC Administrator.

To connect an ODBC client application to a Branch Office database, an application must create a connection to the database. For example,

```
"UID=SYSTEM;PWD=MANAGER;DSN=POLITECL;DATABASE=BRANCH"
```

Table 15–1 describes the above database connection statement.

Table 15–1 Database Connection Statement Description

Parameter	Description
UID	A valid database user.
PWD	A valid password to the database.
DSN	A data source name set up using the ODBC Administrator.
Database	The name of the local Branch Office database residing in the OLDB40 folder in the <i>Oracle_home</i> directory.

15.3.2.2 JDBC Connections

JDBC client applications make connections to the Branch Office database machine as given below.

```
Connect con=Drivermanager.getConnection (JDBC URL,user,password)
```

Table 15–2 describes the above Branch Office database connection statement.

Table 15–2 Branch Office Database Connection Description

Parameter	Description
JDBC URL	The database URL. For example, jdbc:polite@database_host_name:port_number:DSN
User	A valid database user.
Password	A valid password for the database.

Given below is a Java sample that describes connection for multiple users.

```
Connection conn = null

try
{
    Class.forName("oracle.jdbc.driver.OracleDriver");
    conn = DriverManager.getConnection
        ("jdbc:Polite@DATA_SERVER:100:POLITECL", "SYSTEM", "MANAGER");
}
catch(Exception e)
{
    System.out.println("An error has occurred.");
    System.out.println("Error accessing the Multi-user database");
    System.out.println(e);
    System.exit(0);
}
```

The listener service must be started either manually or automatically before network connections can be established. The listener service can be started through the services application in the Control Panel, or through the Branch Office Mobile Manager.

Note: Dynamic port numbers are not supported. You must use port 100.

15.4 Administration

This chapter provides instructions for using the Branch Office Mobile Manager. Topics include:

- [Section 15.4.1, "Overview"](#)
- [Section 15.4.2, "Logging into the Branch Office Manager"](#)
- [Section 15.4.3, "Using the Branch Office Manager"](#)
- [Section 15.4.4, "Managing Branch Office Users"](#)

15.4.1 Overview

The Administration facility is a Web-to-Go browser based application that enables the Branch Office Administrator to monitor and configure Branch Office database services, users, applications, and replication jobs. The Branch Office Administrator is a Web-to-Go user created by the Web-to-Go Administrator and must be included as a member of the Web-to-Go group, "BRANCH ADMINISTRATORS."

The Administration Facility's Database Administration and User Administration modules enable an administrator to perform the following functions:

15.4.1.1 Database Service Administration

This section describes tasks relevant to database service administration. Topics include:

- [Section 15.4.3.4, "Viewing the Status of the Branch Office Database"](#)
- [Section 15.4.4, "Managing Branch Office Users"](#)
- [Section 15.4.3.2, "Starting the Database Service"](#)

- [Section 15.4.3.3, "Stopping the Database Service"](#)

15.4.1.2 Remote User Maintenance

The Administration facility enables user information maintenance capabilities for the Branch Office Administrator to centrally manage user access privileges to the Branch Office database. The Administration facility supports the following user management tasks.

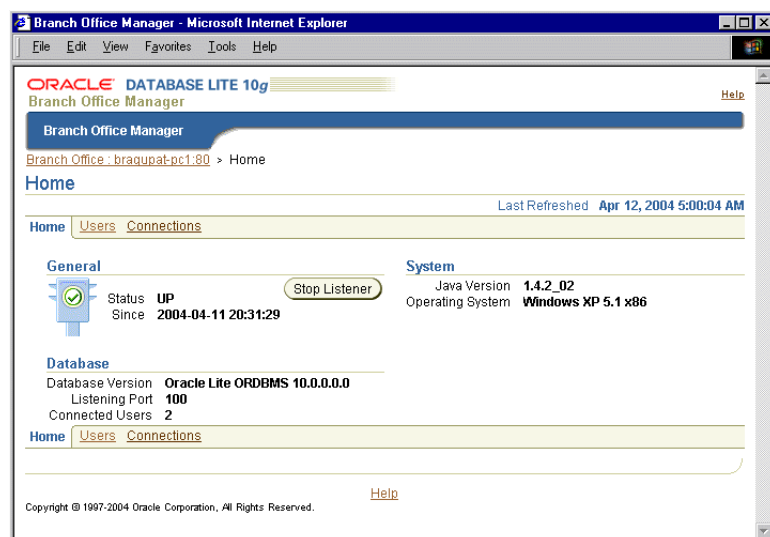
- [Section 15.4.4.1, "Creating Users"](#)
- [Section 15.4.4.2, "Setting User Roles"](#)
- [Section 15.4.4.3, "Setting User Properties"](#)
- [Section 15.4.4.4, "Setting User Privileges"](#)
- [Section 15.4.4.5, "Finding Users"](#)
- [Section 15.4.4.6, "Removing a User"](#)

15.4.2 Logging into the Branch Office Manager

The Branch Office Administrator can access the Branch Office Manager by clicking the Branch Office Manager link in the workspace.

The Branch Office Manager appears and defaults to the Branch Office Home page, as displayed in [Figure 15–6](#).

Figure 15–6 Branch Office Home Page



15.4.3 Using the Branch Office Manager

The Branch Office Manager contains the following pages. It enables the Branch Office Administrator to perform the administrative tasks described below.

- **Home** - The Branch Office home page enables you to start and stop the listener service. It displays general information such as system status, system details such as the java version and operating system, and database information such as version, listening port, and number of connected users.

- Users - The Users page enables you to find and add users to the required database.
- Connections - The Connections page displays connection details such as user name, connection duration, and the database path.

15.4.3.1 Updating Status Summary

The General section of the Branch Office home page provides the listener status to the Branch Office Administrator. The listener status can be changed by starting or stopping the database service. The database section displays the latest status of the Branch Office database. To update the Branch Office status summary, click the Refresh button on your browser. Starting or stopping the listener service also updates the status summary.

15.4.3.2 Starting the Database Service

The Branch Office Manager home page enables the Administrator to start the Windows NT Listener service using the Start Listener button.

15.4.3.3 Stopping the Database Service

The Branch Office home page enables the Administrator to stop the Windows NT Listener service using the Stop Listener button.

Note: A Branch Office Administrator should check the Status Summary for connected users before stopping the service. Local database connections are not detected by the Branch Office Mobile Manager.

15.4.3.4 Viewing the Status of the Branch Office Database

The Branch Office Manager supports an unlimited number of database files. The General and Database section in the Branch Office Manager home page enable an Administrator to view the status of the Branch Office database and start or stop the windows service. The Connection page displays additional database information.

[Table 15–3](#) describes the Branch Office home page.

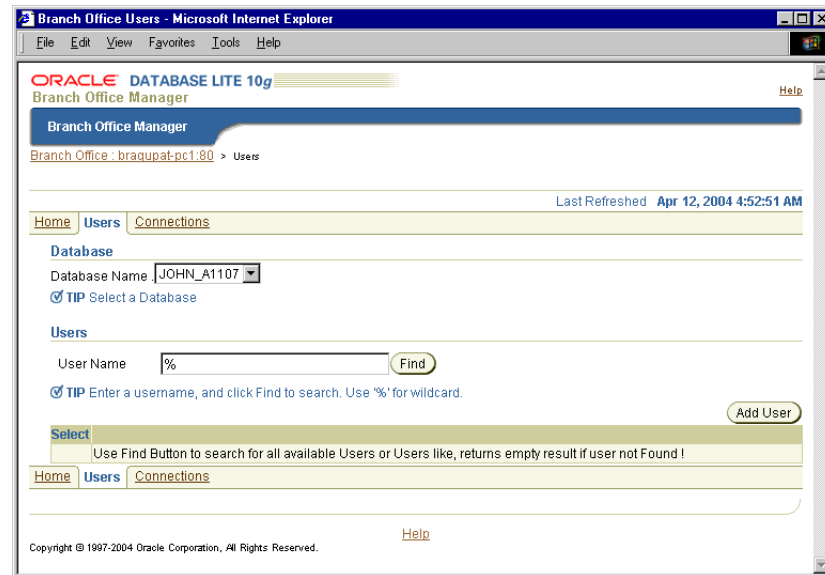
Table 15–3 Branch Office Home Page Description

Label	Function
Status	Branch Office status.
Since	Date and time since the Oracle Database Lite Branch Office system is up.
Database Version	Version number of the Oracle Database Lite Branch Office database.
Listening Port	The server port number that the Oracle Database Lite listener service uses.
Connected Users	Number of currently connected users.
Java Version	Version number of the Java Development Kit.
Operating System	Current operating system.

15.4.4 Managing Branch Office Users

To manage Branch Office users, login to the Mobile Server and navigate to the Users page. As [Figure 15–7](#) displays, the Users page appears.

Figure 15–7 Branch Office Users Page



The Branch Office database does not need to be stopped to administer and maintain users and their access privileges. The Branch Office Administrator can add or delete users while other users are accessing the Branch Office database.

15.4.4.1 Creating Users

To create users, navigate to the Users page and click the Add User button under the Users section. The Add Users page appears. Enter the appropriate data in the corresponding fields and click the Save button.

Note: You should not create a user named "System." This user name is reserved for Web-to-Go use.

15.4.4.2 Setting User Roles

After you create a new user, the Branch Office Mobile Manager automatically displays the Roles page. Using the Roles page, the Branch Office Administrator can assign user roles by selecting the available role boxes. As [Table 15–4](#) describes, the Branch Office Administrator can assign the following roles.

Table 15–4 User Roles Description

Field	Description
DBA	Database administrator privileges. When selected, users can add or remove users and add files to the database.
RESOURCE	RESOURCE privileges. Selecting this check box enables users to create their own sets of tables and relate them to their own schema.

15.4.4.3 Setting User Properties

The Roles Home page enables the Administrator to set user properties. To set user properties, click the Roles Home page link. Using this page, you can modify a user's password.

15.4.4.4 Setting User Privileges

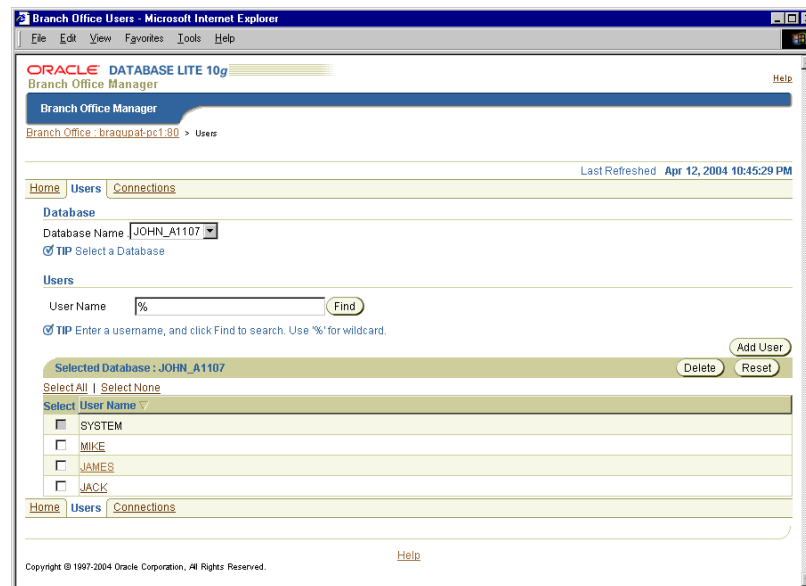
The Privileges page enables the Branch Office Administrator to assign user privileges. To control user access to database tables, you can grant user privileges such as Select, Delete, Insert, and Update.

15.4.4.5 Finding Users

To find all users, click the Users link. Select the appropriate database name and click the Find button. To find a specific user, enter the user name and click the Find button.

To display a list of all users for the chosen database, enter the % sign and click the Find button. As [Figure 15–8](#) displays, the Users page displays users that are associated with the chosen database.

Figure 15–8 *Displaying Branch Office Users*



15.4.4.6 Removing a User

To remove a user, select the check box displayed against a user name and click the Delete button.

15.4.5 Managing Applications

The Applications tab enables the Branch Office Administrator to list all Web-to-Go applications that the Branch Office Administrator can access. Clicking an application link displays a list of files that comprise the application. The Branch Office Administrator can designate certain files as public, which means that they can be viewed and downloaded by the end users.

15.4.5.1 Installing Public Files

To download and install a public file, Branch Office users must access the URL given below.

`http://<branchofficemachine>/public/download`

This URL lists all public files under their respective applications. Users can click the required file name and save it in their file system. After saving the file, users can install the application by running the self-extracting file.

For ODBC configuration, click the ODBC driver link. This downloads the `setup.exe`. After the file is downloaded, users must run the `setup.exe`.

Offline Instantiation

This document discusses the Offline Instantiation feature. Topics include:

- [Section 16.1, "Overview"](#)
- [Section 16.2, "Setting Up the Mobile Server Host and Mobile Development Kit Host"](#)
- [Section 16.3, "Setting Up the Mobile Client Root Directory"](#)
- [Section 16.5, "The OLI CD's Directory"](#)
- [Section 16.6, "The OLI Configuration File: Oli.ini"](#)
- [Section 16.7, "Using the OLI Engine"](#)
- [Section 16.8, "Deploying on Client Machines"](#)

16.1 Overview

The Offline Instantiation (OLI) utility enables mobile administrators to prepare a package which includes the Mobile Client software and initial data for every mobile user in batch mode. In offline mode, this package can be used to set up a Mobile Client with user-specific initial data within Oracle Database Lite. This procedure helps users avoid an expensive online synchronization process as part of the first time Mobile Client setup procedure.

16.2 Setting Up the Mobile Server Host and Mobile Development Kit Host

To set up the Mobile Server host and the Mobile Development Kit (MDK) host, perform the following steps.

1. Install the Mobile Server and Mobile Development Kit.
2. Start the Mobile Server.
3. Create the requisite clients, publications, and subscriptions on the Mobile Server. Subsequent operations such as Mobile Client installation and OLI command execution must be carried out on the MDK host. The MDK host contains a sample OLI configuration file named `oli.ini` and the OLI batch file named `oli.bat` at the following location.

```
<Oracle_home>\Mobile\Sdk\bin
```

16.3 Setting Up the Mobile Client Root Directory

To set up the Mobile Client Root directory, perform the following steps.

-
1. Download `setup.exe` from the Mobile Server setup page, which is available at the following URL.

`http://<mobile_server>:<port>/webtogo/setup`

This step is necessary to establish the correct directory structure and to ensure that the appropriate Oracle Database Lite binaries are used when creating the client bundles for a particular platform.

Choose the appropriate Oracle Database Lite Mobile Client. For example, choose Branch Office for Branch Office applications, Mobile Client for Web-to-Go for web applications, Mobile Client for Web-to-Go (BC4J Support) for web applications with BC4J support, and the Mobile Client for Win32 for Win32 and WinCE applications.

Note: To set up the Mobile Client on the WinCE (Pocket PC) platform, choose the Mobile Client for Win32.

2. Run **setup.exe** and specify an installation directory for the Mobile Client. The install directory specified during this process is defined as `MOBILECLIENT_ROOT` in the OLI configuration file (`oli.ini`). Once the client directory is established, the configuration file should be updated accordingly. All client databases will be instantiated in this directory. Hence, it is important to ensure adequate disk space availability.
3. Ensure that the Mobile Client successfully synchronizes with the Mobile Server. Using the `mSync.exe` application (or the Web-to-Go client if it is the designated platform), initiate the synchronization process for one of the subscribing clients. It may be necessary to uninstall the existing Mobile Client before running the Mobile Client `setup.exe`. Synchronizing before offline instantiation accomplishes two things. The synchronization process validates the setup and downloads any application files that are associated with the subscription that should be included in the client bundle.

16.4 Setting Up the Mobile Client (Distribution) Root Directory

To set up a Mobile Client (Distribution) Root directory, use the same `setup.exe` (used for the client installation), and execute the program on the Command Line as follows.

`setup.exe -d`

The distribution directory specified during this process is defined as `MOBILECLIENT_CD_ROOT` in the OLI configuration file (`oli.ini`). Once the distribution directory is established, the configuration file should be updated accordingly.

Note: Ensure that the `MOBILECLIENT_CD_ROOT` distribution directory and the `MOBILECLIENT_ROOT` directory are different directories.

16.5 The OLI CD's Directory

The OLI CD's directory is a location where all the client packages are placed during the offline instantiation process. This directory must be located on a drive with adequate free disk space for all client databases. In the file `oli.ini`, this directory is called `OLI_CDS_ROOT`.

Note: Ensure that this directory is different from MOBILECLIENT_CD_ROOT and MOBILECLIENT_ROOT as specified in previous steps.

16.6 The OLI Configuration File: Oli.ini

In the file `oli.ini`, you must indicate the location of the Mobile Client Root, Mobile Client CD Root, OLI CD Root, and a list of users requiring offline instantiation to the OLI engine. The sample configuration file is available on the MDK host at the following location.

`<Oracle_home>\Mobile\Sdk\bin\`

The content of this file is given below.

```
#####
#
# OLI.INI
# Oracle 10g Lite Offline Instantiation Configuration File
# Copyright © 1997-2004 Oracle Corporation.
# All Rights Reserved.
#
#####
#
# There are two sections whose names are enclosed in square
# brackets: [SETUP] and [CLIENTS].
# Lines starting with a "#", ";", "--" or "/" are comments.
#
#
# Site specific parameters.
# The format for this section is <PARAMETER> = <VALUE>
#
[SETUP]

#
# The mobile server name or IP. If on a port other than 80, append ":<port>".
# Sync server need be running when OLI is launched.
#
MOBILE_SERVER=hostname.domain:80

#
# If the mobile server port specified above is secure, set "USE_SSL" to "YES".
# Otherwise, use "NO".
#
USE_SSL=NO

#
# The mobile server database repository JDBC URL, mobileadmin schema and password
#
JDBC_URL=jdbc:oracle:thin:@hostname.domain:1521:oid
SCHEMA=MOBILEADMIN_SCHEMA
PASSWORD=MOBILEADMIN_PASSWORD

#
# The method used to populate client databases. Valid values are "SYNC" and
# "JDBC".
# "SYNC": use client-server synchronization to generate ODBs.
# "JDBC": use JDBC to transfer data from server repository to client.
# If clients subscribe to same data for some tables, "JDBC" is faster since they
```

```

are
# transferred only once for all clients.
#
MAKEODB_METHOD=JDBC

#
# The JDBC driver type for the connections to Olite client databases. Valid values
are "NATIVE" and "ODBC".
# "NATIVE": use Olite native driver.
# "ODBC": use SUN JDBC-ODBC bridge.
#
OLITE_JDBC_DRIVER=NATIVE

#
# The destination directory for the Olite mobile client installation.
# This is the directory when you run "setup.exe" without any options.
#
MOBILECLIENT_ROOT=C:\MOBILECLIENT

#
# The directory of the Olite mobile client installation CD.
# This is the destination directory when you run "setup.exe" with an option "-d".
#
MOBILECLIENT_CD_ROOT=C:\MOBILECLIENT_CD

#
# The Directory where OLI puts the client instantiation packages.
# Under this directory, each instantiated client will have a sub directory
# which can be copied to a CD to be used for mobile client installation
# on the client machine. Client ODBs are included.
#
OLI_CDS_ROOT=C:\OLI_CDS

#
# The device type of the targeted mobile client machines.
# Use "WIN32" for win32 native,
# use "WTG" for webtogo client deployments and
# use "WCE" for pocket pc
#
DEVICE_TYPE=WIN32

#
# The number of clients to be processed concurrently
#
THREADS=1

#
# List of clients to be instantiated. The clients must have been created
# on the mobile server.
# The format for this section is <CLIENTID> <PASSWORD>
# Passwords are required
#
[CLIENTS]

CONSC1 MANAGER
CONSC2 MANAGER
CONSC3 MANAGER
CONSC4 MANAGER

```



```

CONSC5 MANAGER
CONSC6 MANAGER
CONSC7 MANAGER
CONSC8 MANAGER
CONSC9 MANAGER
CONSC10 MANAGER
CONSC11 MANAGER
CONSC12 MANAGER
CONSC13 MANAGER
CONSC14 MANAGER
CONSC15 MANAGER
CONSC16 MANAGER
CONSC17 MANAGER
CONSC18 MANAGER
CONSC19 MANAGER
CONSC20 MANAGER

```

16.7 Using the OLI Engine

The OLI engine is a java class named `oracle.lite.sync.OLI_Win32`. It reads the file `oli.ini` in the current directory for information related to configuration settings. It also uses two repository tables which store information related to resuming OLI tasks, during interruptions or failures. These tables are `C$OLI_CLIENTS` and `C$OLI_SETUP`.

Note: Before launching the OLI engine, you must edit the `oli.ini` file. The OLI engine uses two repository tables that store information related to resuming OLI tasks during interruptions or failures. These tables are `C$OLI_CLIENTS` and `C$OLI_SETUP`.

The OLI engine relies on a few java classes and native libraries. To make the java libraries and native libraries accessible to the OLI engine, the software contains a batch file named `oli.bat`, in which the necessary environment variables are set. Using the `oli.bat` file is recommended instead of directly using the class named `oracle.lite.sync.OLI_Win32`.

To launch the OLI engine using the Command Prompt window, locate the directory `<Oracle_home>\Mobile\Sdk\bin` and execute the `oli.bat` file at the Command Line. This action displays the following usage information.

```

Usage
-----
oli.bat [-g] [makeodb] [package] [cleanup] [check status]
        -g  Debug mode

```

To carry out OLI tasks, re-execute the command using the appropriate switches and arguments. The arguments are also called sub-commands.

16.7.1 Using Sub-Commands

The OLI engine provides sub-commands that enable you to create and populate the client database files, create packages for mobile clients, and cleanup OLI tables. As a normal practice, execute them in the given order. The following sub-commands correspond to sub-tasks. These sub-commands, together with the status tables, enable you to resume a sub-task in case of failure or interruption. They are:

-
- [Section 16.7.1.1, "makeodb"](#)
 - [Section 16.7.1.2, "package"](#)
 - [Section 16.7.1.3, "cleanup"](#)

16.7.1.1 makeodb

Creates and populates the client 'odb' files.

Usage

```
oli.bat [-g] makeodb
```

The initial status of a client in the repository table C\$OLI_CLIENTS is RESET. After a client is processed successfully by the sub-command makeodb, its status changes to ODBMADE or SLUG.

Note: Before executing the sub-commands makeodb and package on WinCE or Win32 devices, ensure that you set the DEVICE_TYPE parameter to WCE or Win32 in the oli.ini file.

16.7.1.2 package

The sub-command package creates a package for each client in the directory specified by the parameter OLI_CDS_ROOT in the oli.ini file. Each client's package is contained in a sub-directory, which is named after the client name.

Usage

```
oli.bat [-g] package
```

After a client package is successfully processed by the sub-command package, its status is changed to PACKAGED.

16.7.1.3 cleanup

The sub-command cleanup cleans the OLI tables. To re-create the OLI tables, run the "cleanup" command first. Do not run this command without a need to start again. There is an additional command "checkstatus" which you can run to check the status of OLI clients.

Usage

```
oli.bat [-g] cleanup
```

16.8 Deploying on Client Machines

The deployment process for WinCE applications are different from those of native Win32 applications and Web-to-Go applications.

Deploying Win32 Native and Web-to-Go Applications

To deploy on client machines for native Win32 applications and Web-to-Go applications, perform the following steps.

1. After a successful server side OFF_LINE INSTANTIATION process, each client is provided with a one-click installable package in the directory specified by the parameter named OLI_CDS_ROOT in the oli.ini file. The client's sub directory

(package) is named after the client name. Copy the client's package to the client machine.

2. On the client machine, perform the following:
 - a. Uninstall existing Mobile Client software.
 - b. From the available package, run `setup.exe`.

Deploying WinCE Applications

To deploy on client machines for WinCE applications, perform the following steps.

1. Install the Mobile Client for Windows CE onto the CE device.
2. After a successful server-side `OFF_LINE INSTANTIATION` process, each client contains a package in the directory, which is specified by the parameter `OLI_CDS_ROOT` in the `oli.ini` file. The client's sub directory (package) is named after the client name. Copy the client's package to the `<Oracle_home>` directory of the WinCE device.

Configuring SSL For Mobile Server

Oracle Database Lite 10g supports Secure Socket Layer (SSL) communication between the Mobile Server and Mobile Clients. Oracle Database Lite uses the SSL that is embedded within OC4J, which is shipped as part of Mobile Server.

Note: If you choose to install standalone Mobile Server, the standalone OC4J is installed; however, the other option installs OracleAS as the middle-tier. OracleAS also includes OC4J, but the configuration for SSL is more involved. This chapter covers the basic SSL configuration for the standalone Mobile Server. See the *Oracle Application Server Containers for J2EE Security Guide* for more information on all aspects of configuring SSL.

This chapter assumes that you understand the concepts behind SSL and provides only the steps for using keys and certificates for SSL communication for the standalone Mobile Server.

- [Creating an SSL Certificate](#)
- [Configuring Mobile Server for SSL](#)

Note: These are server-level steps which are typically executed prior to deployment of an application that requires SSL communication.

17.1 Creating an SSL Certificate

Oracle Database Lite ships a sample keystore file with a self-signed sample certificate. Use this keystore only for development or testing purposes. Obtain a signature from a recognized certificate authority for all production systems. The test keystore is located in the following directory:

`ORACLE_HOME\Mobile\Server\Bin\samplekeystore`

To create a keystore file, perform the following steps:

1. Use the Sun Microsystems Java `keytool` utility to generate a private key, public key, and an unsigned certificate. Place this information into either a new or existing keystore.

Note: A keystore is a `java.security.KeyStore` instance that you create and manipulate using the `keytool` utility, which is provided with the Sun Microsystems JDK. See <http://java.sun.com/j2se/1.3/docs/toolbox> for more information on the `keytool` utility.

2. Obtain a signature for the certificate, using either of the following approaches:
 - Generate your own signature by using `keytool` to self-sign the certificate. This is appropriate only if your clients trust you as your own certificate authority.
 - Obtain a signature from a recognized certificate authority through the following steps:
 - a. Using the certificate from Step 1, use `keytool` to generate a certificate request, which requests a certificate authority to sign the certificate.
 - b. Submit the certificate request to a certificate authority.
 - c. Receive the signature from the certificate authority and import it into the keystore using `keytool`. In the keystore, the signature is matched with the associated certificate.

Each certificate authority has its own process for requesting and receiving signatures. Since this is outside the scope and control of Oracle Database Lite, it is not covered in Oracle Database Lite documentation. However, the SSL section in the *Oracle Application Server Containers for J2EE Security Guide* has an example of how to generate your own keystore. For other information, go to the Web site of any certificate authority. Each browser lists trusted certificate authorities.

Here are the Web addresses for VeriSign, Inc. and Thawte, for example:

<http://www.verisign.com/>

<http://www.thawte.com/>

17.2 Configuring Mobile Server for SSL

Once you have a certificate, you must configure SSL in the application server that is installed with the Mobile Server. When you installed, you chose to install the Mobile Server either in standalone mode or to use the application server. Both of these environments are discussed below:

- [Configuring SSL for Mobile Server With Oracle9iAS or OracleAS](#)
- [Configuring SSL for Standalone Mobile Server](#)

17.2.1 Configuring SSL for Mobile Server With Oracle9iAS or OracleAS

For production systems, you have either Oracle9iAS or OracleAS installed with the Mobile Server. You must configure SSL on both the application server and the Mobile Server, as follows:

1. Configure SSL in the application server using the administration GUI. The directions on how to configure SSL when using Oracle9iAS or OracleAS as your middle-tier is in the SSL or HTTPS chapter in the *Oracle Application Server Containers for J2EE Security Guide*.

2. Configure SSL in the Mobile Server by adding `SSL=YES` in the `[WEBTOGO]` section of the `ORACLE_HOME\Mobile\Server\bin\webtogo.ora` file.
3. After all configuration is complete, restart the application server to initialize the changes.

17.2.2 Configuring SSL for Standalone Mobile Server

With the standalone Mobile Server, the standalone version of the OC4J application server is installed with the Mobile Server. To configure SSL for this environment, you modify the Mobile Server `webtogo.ora` file and certain XML elements within the OC4J XML configuration files, as follows:

1. Configure SSL in the Mobile Server by adding `SSL=YES` in the `[WEBTOGO]` section of the `ORACLE_HOME\Mobile\Server\bin\webtogo.ora` file.
2. If you do not have a `secure-web-site.xml` file, then copy and rename the `default-web-site.xml` to `J2EE_HOME\config\secure-web-site.xml`.
3. Edit the `secure-web-site.xml` file with the following elements:

- a. Add `secure="true"` to the `<web-site>` element, as follows:

```
<web-site port="443" display-name="Oracle Application Server Containers for
J2EE Web Site" secure="true">
```

- b. Add the following new line inside the `<web-site>` element to define the keystore and the password:

```
<ssl-config keystore="YourKeystore" keystore-password="YourPassword" />
```

where *YourKeystore* is the path and name of the keystore and *YourPassword* is the keystore password. The path for the keystore can either be a full path or a path that is relative to `J2EE_HOME\config`. In addition, you can hide the password through password indirection. This is discussed fully in the *Oracle Application Server Containers for J2EE Security Guide*. For example, with a keystore of `../../keystore` and password of `"oracle"`, the configuration is as follows:

```
<!-- Enable SSL -->
<ssl-config keystore="../../keystore" keystore-password="oracle"/>
```

- c. Change the `<web-site>` element port number to use an available port. The reason you must change the port is because you copied this file from `default-web-site.xml`, which uses the port that is currently configured. Thus, choose a port that can be used for SSL communication; for example, the default for SSL ports is 443.
- d. Save the changes to `secure-web-site.xml`.
4. Edit the `server.xml` file to point to the `secure-web-site.xml` file.
 - a. Uncomment or add the following line in the file `server.xml` so that the `secure-web-site.xml` file is added to the OC4J initialization.


```
<web-site path="./secure-web-site.xml" />
```
 - b. Save the changes to the `server.xml` file.
5. Stop and re-start OC4J to include the `secure-web-site.xml` file modifications.
6. Test the SSL port by accessing the Mobile Server in a browser on the SSL port. For example, `https://<yourserver>:443/webtogo`.

If you are using the test keystore file or your own self-signed certificate, you will be asked to accept the certificate, since the SSL certificate used is not signed by an accepted certificate authority. When completed, Mobile Server listens for SSL requests on the port configured in the `secure-web-site.xml` file and listens for non-SSL requests on the port configured in the `default-web-site.xml` file. You can disable either SSL requests or non-SSL requests, by commenting out the appropriate `*web-site.xml` in the `server.xml` configuration file.

```
<web-site path="./secure-web-site.xml" /> - comment out this to remove SSL
<default-site path="./default-web-site.xml" /> - comment out this to remove
non-SSL
```

17.3 Troubleshooting Error Messages for an SSL-Enabled Mobile Server

The following errors may occur when using SSL certificates on your Mobile Server:

No available certificate corresponds to the SSL cipher suites which are enabled

Cause: Something is wrong with your certificate.

Action: Examine your certificates and check that at least one of them supports the SSL cipher suite you are using.

IllegalArgumentException: Mixing secure and non-secure sites on the same ip + port

Cause: You cannot configure SSL and non-SSL web-sites to listen on the same port and IP address.

Action: Check to see that different ports are assigned within `secure-web-site.xml` and `default-web-site.xml` files.

Troubleshooting FAQ's

This document contains frequently asked questions for troubleshooting the Mobile Server. Topics include:

- [Section A.1, "Inspecting Files in the Mobile Server Repository"](#)
- [Section A.2, "Running the Mobile Server in Debug Mode"](#)
- [Section A.3, "First Synchronization Causes Browser to Timeout"](#)
- [Section A.4, "Accessing the Client Database Offline"](#)
- [Section A.5, "Accessing Schema Changes to a Published Application"](#)

A.1 Inspecting Files in the Mobile Server Repository

You can use the Mobile Server shell utility (wsh) to inspect and modify the Mobile Server Repository interactively. Start the Command Prompt and enter the following.

```
wsh -L system/x@olite-db
```

OR

```
wsh -o <adminuser>/<adminuser's password>@o8db
```

For example, you could enter the following sample codes at the command prompt.

```
wsh -o administrator/admin@webtogo.world  
wsh -L system/x@webtogo
```

This displays the Mobile Server Repository prompt.

The following table lists commands that are available for inspecting and altering the Mobile Server Repository.

[Table A-1](#) describes available commands for inspecting and altering the Mobile Server Repository.

Table A-1 *Commands to Inspect and Alter the Mobile Server Repository*

Command	Definition
dir	Displays a list of files in a directory.
copy	Copies one or more files to another location.
cp	Copies one or more files to another location.
edit	Launches Notepad for editing a file.
del	Deletes one or more files.

Table A-1 (Cont.) Commands to Inspect and Alter the Mobile Server Repository

Command	Definition
rm	Deletes one or more files.
cd	Displays the name or changes the current directory.
md	Creates a directory.
rd	Removes (deletes) a directory. Use the option <code>-s</code> to remove a directory including all subdirectories.
type	Displays the contents of a text file or files.
exit	Quits the command shell.
quit	Quits the command shell.
help	Provides help information for shell commands.
sync	Synchronizes the file system with the database.

A.2 Running the Mobile Server in Debug Mode

If you experience any difficulty with the Mobile Server running with the Oracle9iAS or the standalone Mobile Server, you can run the Mobile Server in debug mode.

To run the Mobile Server in debug mode, perform the following steps.

1. Shut down the Mobile Server and Oracle9iAS or the standalone Mobile Server.
2. Verify in the file **webtogo.ora** that the following [DEBUG] parameters contain the specified values.

Table A.2 describes Debug parameters that are available in the webtogo.ora file.

Table A-2 Debug Parameters

Parameter Name	Value
TRACE_ENABLE	YES
TRACE_REMOTE_MACHINE	<debug_client> where <debug_client> is the name of the machine to which you want Oracle9iAS to send debugging messages.
TRACE_REMOTE_PORT	100

3. Start the Command Prompt on the Debug machine and enter the following command.

```
wsh -m 100
```

In the above command, `wsh` is the application, `-m` enables the monitoring feature, and `100` specifies the port to monitor.

4. Restart the Mobile Server.

Note: If you are running the standalone Mobile Server instead of the Oracle9iAS, you can debug your Mobile Server by specifying `-d0` on the Command Line. For example, `webtogo -d0`.

A.3 First Synchronization Causes Browser to Timeout

The Problem

The duration of the first synchronization process, between the client and the server may take a very long time (For example, upwards of 45 minutes), causing Microsoft's Internet Explorer browser to time out.

The Solution

This solution is available only for Microsoft's Internet Explorer. For the Mobile Client for Web-to-Go, change the `ReceiveTimeout` value for a particular registry key on Windows 32. The following paragraphs provide the location of this parameter and specifies how to change its value.

1. If you want to change the `ReceiveTimeout` value (that is, the number of milliseconds that the browser will wait to receive the data from the server), uncomment the following two lines in the **REGISTRY** section of the file **setup.ini**. This file is downloaded to the client from the server when the Mobile Client for Web-to-Go is first installed on the client machine.

```
#KEY: HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\
      Internet Settings
#VALUEDWORD: ReceiveTimeout = 40000000
```

2. To uncomment the two lines, remove the hash marks in front of the **KEY** and **VALUEDWORD** statements and then change the `ReceiveTimeout` value to the desired value in milliseconds. In the example given below, the timeout value is 40000 seconds.

```
wsh -o mobileadmin/manager@webtogo.world
cd setup
edit setup.ini
```

3. When you modify the file **setup.ini** in the Mobile Server Repository, the next time that the file is downloaded to the Mobile Client, these two lines will be uncommented.

A.4 Accessing the Client Database Offline

When you go offline, after being online, and you need to access your client database, use your mobile user's password, not the password manager. It is the password that you enter when you logged in, before going offline.

If you do not use your mobile user's password when you try to access your client database offline, you will receive the following error message.

```
[POL-5150] access violation
```

For example, if you log in as JOHN/JOHN on a Windows 32 machine with contains a Mobile Client for Web-to-Go, and JOHN goes offline, the user database created for JOHN on the client machine requires JOHN's password for access.

If you use **msql** to access the user database, start the Command Prompt and enter the following statement.

```
msql system/john@jdbc:polite:john_todo
```

In this example, JOHN's password is john, and the DSN name that was created automatically on the client for the user name JOHN is `john_todo`.

Syntax

The following syntax enables you to access the user database.

```
msql system/<mobile user's password>@jdbc:polite:<dsn created for the user>
```

To find the DSN name, use the ODBC Admin tool on your client machine.

A.5 Accessing Schema Changes to a Published Application

To include the schema evolution changes in your existing application, you must republish the application. If one or more of the application tables has changed and you want these changes to be reflected on the client, you must republish the application using the Packaging Wizard. For more information on how to publish your applications, see the *Oracle Database Lite Developer's Guide*.

Mobile Server Configuration Parameters

This document describes configuration parameters for the Mobile Server. These parameters are included in the file `webtogo.ora`. The Mobile Server uses the `webtogo.ora` file to initialize the Mobile Server. When you launch the Mobile Server, it reads the parameters in the `webtogo.ora` file. This document defines the following system-wide parameters for the Mobile Server. Topics include:

- [Section B.1, "\[WEBTOGO\]"](#)
- [Section B.2, "\[FILESYSTEM\]"](#)
- [Section B.3, "\[DEBUG\]"](#)
- [Section B.4, "\[PUBLIC\]"](#)
- [Section B.5, "\[SERVLET_PARAMETERS\]"](#)
- [Section B.6, "\[CONSOLIDATOR\]"](#)

B.1 [WEBTOGO]

The following WEBTOGO parameters control the behavior of both the Mobile Client for Web-to-Go and the Mobile Server.

[Table B-1](#) lists WEBTOGO parameters and their usage definitions.

Table B-1 WEBTOGO Parameters

Parameter	Definition
<code>USE_SYSTEM_CLASSPATH=YES</code>	If set to yes, searches for Java classes in the computer's classpath before searching the Mobile Server Repository.
<code>MODE=SERVER</code>	The mode the Mobile Server is running in. Valid modes are <code>SERVER</code> , <code>CLIENT</code> , and <code>BRANCH</code> . The value <code>BRANCH</code> indicates that the Mobile Server is running in <code>BRANCH</code> mode for client operations.
<code>PORT=80</code>	The port number on which the Mobile Server is running. Not valid in Oracle9i Application Server (Oracle9iAS) installation.
<code>CLASSPATH=</code>	A list of directories in the Mobile Server Repository. Web-to-Go searches for Java classes.
<code>PROXY_SERVER=proxy.com</code>	The proxy host name and number. The Mobile Client for Web-to-Go setup modifies this entry.
<code>PROXY_PORT=80</code>	The proxy port number. The Mobile Client for Web-to-Go setup modifies this entry.
<code>SQL_RETRIES=5</code>	Number of attempts to modify a JDBC connection before timing out.

Table B-1 (Cont.) WEBTOGO Parameters

Parameter	Definition
SESSION_EXPIRATION=45	Logon session expiration time. The system logs out a user after the specified minutes of inactivity.
KEY_EXPIRATION=20	Number of minutes elapsed before invalidation.
PUBLIC_NAME=/public	The public URLs name. The default value is /public.
BASE_URL=/webtogo	Base URL on which Web-to-Go is installed in the Oracle9i Application Server (Oracle9iAS).
ADMIN_PORT=8080	Admin port for starting the Mobile Server.
ADMIN_TNS_NAME=WEBTOGO.WORLD	The Mobile Server Repository's TNS name.
ADMIN_JDBC_URL=jdbc:oracle:oci8@webtogo.world	Mobile Server Repository's JDBC URL.
FONT_NAME=Arial	The Web-to-Go Workspace font.
APPLET_USE_THIN_JDBC=YES	Requests that JDBC use the thin driver or the Web-to-Go data communication link for all database calls. Web-to-Go uses the internal Web-to-Go JDBC driver, if it is not using the JDBC thin driver. If this parameter is set to YES, then the parameter THIN_JDBC_URL should also be set.
THIN_JDBC_URL=jdbc:oracle:thin:@foo-pc:1521:orcl	The Mobile Server Repository's thin JDBC URL. This URL is used by the JDBC thin driver to connect to the Mobile Server Repository database.
LOAD_LIBRARIES	The list of system libraries (DLLs) that need to be preloaded by the Mobile Server, separated by semi-colons. For example, LOAD_LIBRARIES=myapp;olmuadm. The DLLs myapp.dll and olmuadm.dll is loaded when the Mobile Server is started.
SOCKET_TIMEOUT	The number of seconds that a read() call on the server socket will block before timing out. When set to 0, the read call will never time out. The default value is 60 seconds.
ADMIN_USER	Encrypted user name. Users must not try to edit the encrypted user name. This parameter can be set by navigating to the following URL. <server>/webtogo/startup
ADMIN_PASSWORD	Encrypted user password. Users must not try to edit the encrypted password. This parameter can be set by navigating to the following URL. <server>/webtogo/startup
RESTRICTED_ADMIN_HOSTS=<list of comma separated IP addresses>	This parameter provides security for accounts with Administrator access. With this parameter, the Mobile Server can be configured to allow login requests to a specified set of IP addresses for accounts with Administrator access. With this parameter, you can also restrict access to the Mobile Server Startup feature. Only valid login requests from a browser that runs on machines whose IP address is listed as a value of this parameter will be granted access. For example, RESTRICTED_ADMIN_HOSTS=144.125.127.150.144.125.127.101 Note: Users who have Administrator access should not connect through a proxy server.

Table B-1 (Cont.) WEBTOGO Parameters

Parameter	Definition
SSL=YES	If this parameter is set to YES, then the Mobile Server runs in SSL mode. To use this feature, the Mobile Server should be running as a module inside Oracle9i Application Server (Oracle9iAS).
CUSTOM_WORKSPACE=no	Indicates whether or not a custom workspace should be used.
CUSTOM_DIRECTORY=/myworkspace	Location of the custom workspace files in the repository.
DEFAULT_PAGE=myfirstpage.html	The first page of the custom workspace. This page appears when the user accesses the following URL. http://<server>/webtogo
CUSTOM_FIRSTSERVLET=HelloWorld;/hello	Use this parameter to add the first servlet to the custom workspace. Within the first servlet, you can add more servlets to the custom workspace, using the addServlet() call. Format: class;virtual path
WTG_PROXY	HTTP proxy used to connect to the Mobile Server for application deployment. Sample Value: www-proxy.dlsun1.com
WTG_PROXY_PORT	HTTP proxy port used to connect to the Mobile Server for application deployment. Sample Value: 80
TRACE	Enables debug output. Sample Value: NO
TRACE_ALL	Enables detailed debug output. Includes SQL statements and execution times. Sample Value: NO
JAVA_OPTION=value	Use this parameter to specify additional runtime arguments for the Java VM. Example: JAVA_OPTION=-Djbo.SQLBuilder=OLite -Djbo,TypeMapEntries=Oracle
DEFAULT_CLIENT_1CLICK	The default value for the Mobile Client's "use default setting for sync" Sample Value: YES
DEFAULT_CLIENT_UPGRADE	The default value for the Mobile Client's "ask before upgrade" setting. Sample Value: YES
DEFAULT_CLIENT_SYNCONLY	The default value for the Mobile Client's "offline only/online/offline" setting. Sample Value: YES
APPLET_SUPPORT_ENABLE=YES	If you want to run an applet that uses a JDBC connection on the Mobile Client for Web-to-Go, you must set this parameter to YES and restart the client. If the applet does not use a JDBC connection, you need not set this parameter. Setting this parameter to YES for an applet that does not use a JDBC connection, does not impair your settings.

Table B–1 (Cont.) WEBTOGO Parameters

Parameter	Definition
SERVER_URL=http://<mobile_server_name>:<port_number>/webtogo	<p>This parameter points to the Mobile Server. It communicates with the Mobile Server over HTTP or HTTPS. Usually, you need not modify this parameter. If you want to run the Mobile Client for Web-to-Go and download the Mobile Client for Web-to-Go from the following URL, <code>https://<mobile_server_name>/setup</code>, the Mobile Client for Web-to-Go is automatically configured for SSL, and no manual configuration is required. The Mobile Client communicates with the Mobile Server over SSL.</p> <p>However, if you do not download the Mobile Client for Web-to-Go from the Mobile Server that is running in SSL mode and you want to run your Mobile Server in SSL mode, you must modify the <code>SERVER_URL</code> parameter in the configuration file <code>webtogo.ora</code>, on the client side as displayed in the left column.</p>
SYNC_CANCEL	<p>This parameter can be set on the client side to determine if the "Cancel" link should appear on the synchronization page.</p> <p>If this parameter is set to <code>YES</code>, the "Cancel" link appears on the synchronization page. By clicking the Cancel link, you can stop the data synchronization. The link will not appear after the data synchronization is complete.</p>
MAX_THREAD_POOL	Limits the number of threads available in the connection pool. If threading problems occur, set this parameter to 0 or 1.
IAS_MODE	<p>This parameter must be set to the value <code>YES</code> only if the Mobile Server is running as a component of Oracle9iAS.</p> <p>Example: <code>IAS_MODE=YES</code></p> <p>If the Mobile Server is running in Standalone mode or as a component of Oracle9iAS 1.0.2.2.0, this parameter must be set to the value <code>NO</code>.</p> <p>The default value is <code>NO</code>.</p>

B.2 [FILESYSTEM]

The following FILESYSTEM parameters control the behavior of the Mobile Server Repository.

Table B–2 lists [FILESYSTEM] parameters and their definitions.

Table B–2 FILESYSTEM Parameters

Parameter	Definition
TYPE	<p>Type of File system.</p> <p>O8 - Oracle based file system.</p> <p>OL - Oracle Lite based file system.</p> <p>OS - Operating system's file system.</p> <p>MIXED - Mixed file system.</p>
PRIMARY=OL	Primary file system in MIXED mode.
SECONDARY=OS	Secondary file system in MIXED mode.

Table B–2 (Cont.) FILESYSTEM Parameters

Parameter	Definition
ROOT_DIR=<ORACLE_HOME>/MOBILE/SERVER/REPOSITORY	Root Directory. Valid only for OS file system. This directory path format applies to the environment where the Mobile Server runs on Solaris. Replace <ORACLE_HOME> with your actual Oracle Home.
ROOT_DIR=<ORACLE_HOME>\MOBILE\SERVER\REPOSITORY	Root directory. Valid only for OS file system. This directory path format applies to the environment where the Mobile Server runs on Windows NT. Replace <ORACLE_HOME> with your actual home.
CACHE_SIZE=10	08 file cache system in MB. Note: Do not use this flag unless you have sufficient memory in your system.
MAX_CONNECTIONS=1	Maximum number of database connections for the 08 file system.

B.3 [DEBUG]

The following DEBUG parameters control the debugging messages in the Mobile Server.

Table B–3 lists DEBUG parameters and their definitions.

Table B–3 DEBUG Parameters

Parameter Name	Definition
TRACE_ENABLE	Used to turn the trace feature on or off. When the Trace feature is off, trace output is not generated. This value is only overridden when the Mobile Server is running in Standalone mode and with the -d0 command line option on. For example: TRACE_ENABLE=NO Is overridden by -d0 and the trace output is generated to the Console instead of being generated to a file. Sample Value: YES

Table B–3 (Cont.) DEBUG Parameters

Parameter Name	Definition
TRACE_DESTINATION	<p>Trace destinations are Console, File, and Remote. The Administrator can set this parameter to any of these destinations. The Console option generates trace output to the console screen.</p> <p>Note: This trace destination is available only when the Mobile Server is running in Standalone mode. If you set this parameter to the option -d0, the trace output appears on your Console window without appearing in a file, because using the -d0 option with this parameter overrides the trace settings for other trace parameters, such as destination and level, in the webtogo.ora file. The -d0 setting enforces the trace output to appear on your console screen instead of appearing in a file.</p> <p>The File option generates trace output to a file. For more information, see TRACE_FILE_NAME, TRACE_FILE_SIZE, and TRACE_FILE_POOL_SIZE.</p> <p>The Remote option generates trace output to a remote machine where it can be viewed using <code>wsh -m%TRACE_REMOTE_PORT%</code>. See also TRACE_REMOTE_MACHINE, TRACE_REMOTE_PORT.</p> <p>Sample Value: TRACE_DESTINATION=FILE</p>
TRACE_REMOTE_MACHINE	<p>Machine name where <code>wsh -m</code> is running. The Mobile Server sends the debug output to the machine where <code>wsh -m</code> is running.</p>
TRACE_REMOTE_PORT=2080	<p>Trace output is generated to the named port on the %TRACE_REMOTE_MACHINE% on which the trace output can be viewed using <code>wsh -m TRACE_REMOTE_PORT%</code>. See also TRACE_REMOTE_MACHINE.</p> <p>Sample Value: TRACE_REMOTE_HOST=2080</p>
TRACE_FILE_NAME=trace.log	<p>Used as base name to arrange trace files in sequential order starting from 1 to FILE_TRACE_POOL_SIZE.</p> <p>For example: If you set the following parameters.</p> <p>TRACE_FILE_NAME=mytrace.log</p> <p>TRACE_FILE_POOL_COUNT=5</p> <p>then, the Trace files will be named mytrace1.log, mytrace2.log, mytrace3.log, mytrace4.log, mytrace5.log, based on how you set the TRACE_FILE_PER_USER parameter.</p> <p>Sample Value: trace.log</p>

Table B-3 (Cont.) DEBUG Parameters

Parameter Name	Definition
TRACE_LEVEL=1	<p>There are three levels of trace messages:</p> <p>1 (binary 00000001), Basic Trace: General system information, most of the Web-To-Go trace output belongs to this level.</p> <p>2 (binary 00000010), Function Trace: Traces the function sequence being called, mostly used by Consolidator.</p> <p>4 (binary 00000100), SQL Trace: Traces SQL queries being executed, mostly used by Consolidator.</p> <p>In addition, all errors and exceptions are sent to level -1, which have the binary 11111111. All java System.out output are sent to level 9. Both these two levels are always generated as output, if the user is not filtered out. For more information, see TRACE_USER.</p> <p>The parameter value for TRACE_LEVEL is used to do a Bitwise AND operation against all 3 trace levels. If the result is greater than 0, then trace output of that level will be generated as trace output.</p> <p>The parameter value for TRACE_LEVEL used to do a Bitwise AND operation against all 3 trace levels. If the result is greater than 0, then trace output of that level will be generated as trace output.</p> <p>EXAMPLE: If you set the following parameters, TRACE_LEVEL=3, then the Basic and SQL level trace output is generated, but not Function level trace as the & character is a Bitwise AND operator.</p> <p>3 & 1 (Basic) = 1 > 0 3 & 2 (SQL) = 2 > 0 3 & 4 (Function) = 0 = 0</p>
TRACE_USERS	<p>List of valid user names. The user trace and system trace information which is listed is generated as trace output.</p> <p>If the value is an empty string "", then every user is traced. If the value is or contains TRACE_NO_USER, then no actual user is traced. Only the system trace information is generated as trace output.</p> <p>Note: As the administrator, you must not use the TRACE_NO_USER value as the user name.</p> <p>Example: If you set this parameter as follows, TRACE_USERS=jane,jack, then only jane and jack's trace information is generated and displayed as trace output.</p>

Table B–3 (Cont.) DEBUG Parameters

Parameter Name	Definition
TRACE_FILE_PER_USER=YES	<p>Used to specify an individual trace file pool for every individual user. Applicable only when the File option is the Trace destination.</p> <p>If set to YES, then every traceable user has an own trace file pool, and the trace file name includes the user's name. In addition, the system trace output goes to the user's system trace file.</p> <p>If set to NO, all traceable users share the same trace file pool, the actual trace file does not contain any user name.</p> <p>Example: TRACE_FILE_POOL_PER_USER=No</p>
TRACE_FILE_SIZE=10	<p>Used as the maximum file size in MB for trace files. If the threshold value is about to be reached, the trace feature generates output to the next trace file in the pool. For more information, see TRACE_FILE_NAME, TRACE_FILE_POOL_SIZE, and TRACE_FILE_PER_USER.</p>
TRACE_FILE_POOL_SIZE=5	<p>The default value is 5. This parameter specifies the number of files in the trace file pool. If the pool limit is reached, the trace output is overwritten to the first file in the pool. See also TRACE_FILE_NAME, TRACE_FILE_POOL_SIZE, and TRACE_FILE_PER_USER</p>
TRACE_REMOTE_HOST=<local_host>	<p>The trace output is generated to the named machine, on which the synchronized trace can be viewed using the <code>wsh -m%TRACE_REMOTE_PORT%</code>. See also TRACE_REMOTE_PORT.</p> <p>Example: TRACE_REMOTE_HOST=adminhost</p>

B.4 [PUBLIC]

The following PUBLIC parameters control public availability of servlets in the Mobile Server. To make a servlet public, you can use the parameters as listed in the following table.

Table B–4 lists PUBLIC parameters and their definitions.

Table B–4 PUBLIC Parameters

Parameter Name	Definition
myservlet=/ <virtualpath>	<p>To call this public URL from your application, call it as follows:</p> <p><code>http://<server>/public/<virtual path></code></p> <p>For example, <code>oracle.codeMyservlet=/my servlet</code></p> <p><code>oracle.codeMyservlet=/myservlet</code></p>

B.5 [SERVLET_PARAMETERS]

In the SERVLET parameters section, you can list the set of custom parameters which are available to all servlets inside the Mobile Server.

Table B–5 lists SERVLET_PARAMETERS and their definitions.

Table B–5 *SERVLET Parameters*

Parameter Name	Definition
MY_VAR=MY_VALUE	Custom parameter which can be accessed by all servlets.

B.6 [CONSOLIDATOR]

The CONSOLIDATOR parameters control the behavior of the Consolidator. The values that are listed in the following table are default values.

Starting with the Oracle Database Lite 10g Edition, the Consolidator component uses a new log engine that supports new parameters for logging. They are:

- GLOBALLogger
- SYNCLogger
- MGPLLogger
- MGPAPPLYLogger
- MGPCOMPOSELogger

Each parameter sets up a logger for a component which you can use to specify the trace level, trace type, trace destination, trace file pool size, trace file size, and trace users in the following sample format.

```
XLogger=TRACE_LEVEL=<trace_level>|TRACE_TYPE=<trace_type[, trace_type...]>|TRACE_DESTINATION=<trace_destination>[|TRACE_FILE_POOL_SIZE=<trace_file_pool_size>|TRACE_FILE_SIZE=<trace_file_size>|TRACE_USER=<trace_users>]
```

The following parameters have been made part of the above listed logging parameters.

- TRACE_LEVEL
- TRACE_DESTINATION
- TRACE_FILE_POOL_SIZE
- TRACE_FILE_SIZE
- TRACE_USERS

Although the meaning of some of these parameters remains the same, the acceptable values are different. [Table B–6](#) describes parameter values that are acceptable.

Table B–6 *Acceptable Parameter Values*

Parameter Name	Acceptable Values
TRACE_DESTINATION	LOCAL_CONSOLE, TEXTFILE
TRACE_LEVEL	MANDATORY, WARNING, NORMAL, INFO, CONFIG, FINEST, ALL

The parameters TRACE_FILE_POOL_SIZE and TRACE_FILE_SIZE are only applicable for the GLOBALLogger only.

The new log engine does not support the parameters that have been used in the old log engine. They are:

- TRACE_ENABLE
- TRACE_REMOTE_PORT

- TRACE_REMOTE_MACHINE
- TRACE_FILE_PER_USER
- TRACE_FILE_NAME
- TRACE_REMOTE_HOST

Table B–7 lists CONSOLIDATOR parameters and their definitions.

Table B–7 Consolidator Parameters

Parameter Name	Definition
MAX_THREADS=3	Specifies the number of threads spawned within the MGP process. This parameter value should be set to an equivalent number of CPUs.
TEMP = C:\TEMP	Specifies the directory where the trace file is written. The log files saved in the TEMP directory are for the Resume Type (Reliable Transport) only.
MAX_CONNECTIONS=1000	Sets the maximum number of JDBC connections that can be open at one time by the Mobile Server. When this number is reached, no further synchronization sessions are allowed until active connections are released back to the connection pool.
MAX_CONCURRENT	Sets the maximum number of concurrent active synchronization sessions. When this number is reached, subsequent synchronization session requests are placed in a FIFO queue, and are only allowed to continue when active sessions complete and slots become available. This parameter is designed to prevent the Mobile Server from being overloaded by concurrency and can help to improve throughput when the hardware is being stressed.
SLEEP_TIME=20000	Specifies how long (in milliseconds) the MGP sleeps between client synchronization processes.
CONNECTION_POOL=YES	Enables pooling of database connections.
CONNECTION_TIMEOUT=120	Specifies in minutes the JDBC connection timeout for the synchronization session.
COMPOSE_TIMEOUT=300	Specifies in seconds the MGP timeout for a client's synchronization process to complete. If the synchronization does not complete, MGP will retry Compose in the next cycle.
TRACE_DESTINATION	The Administrator can set this parameter to any of these destinations: LOCAL_CONSOLE or TEXTFILE. The Console option generates trace output to the Console screen. The TEXTFILE option generates trace output to a file. See also TRACE_FILE_SIZE, and TRACE_FILE_POOL_SIZE. Sample Value: TRACE_DESTINATION=TEXTFILE

Table B-7 (Cont.) Consolidator Parameters

Parameter Name	Definition
TRACE_LEVEL	<p>Trace Level parameter can be set to the following trace message levels:</p> <p>MANDATORY: This option logs mandatory messages only. For example, Program Exceptions. Regardless of component settings, this option logs exceptions in the error log file (<code>err.log</code>) located in the <code>Conslog</code> directory.</p> <p>WARNING: This option logs warning messages and messages at the Mandatory level. For example, Program Exceptions that users can ignore, messages that the program wants to warn the users with, and so on.</p> <p>NORMAL: This option logs normal messages that the user must be informed with and messages at the Mandatory and Warning level.</p> <p>INFO: This option logs information messages and messages at the Mandatory, Warning, and Normal levels.</p> <p>Examples:</p> <ul style="list-style-type: none">■ Timing of synchronization: When the <code>SYNCLogger</code> is set to the <code>TRACE_TYPE=TIMING</code> and <code>TRACE_LEVEL=INFO</code>.■ MGP Apply: When the <code>MGPAPPLYLogger</code> is set to the <code>TRACE_TYPE=TIMING</code> and <code>TRACE_LEVEL=INFO</code>. MGP Apply must be started with <code>Timing of</code>. <code>COMPOSE</code> must be started with <code>Timing of MGP Compose</code>. MGP must be started with <code>Timing of</code>.■ COMPOSE: When the <code>MGPAPPLYLogger</code> is set to the <code>TRACE_TYPE=TIMING</code> and <code>TRACE_LEVEL=INFO</code>.■ Status of MGP: When the <code>MGPLogger</code> is set to the <code>TRACE_TYPE=GENERAL</code> and <code>TRACE_LEVEL=INFO</code>. <p>CONFIG: This option logs configuration messages and messages at the Mandatory, Warning, Normal, and Info levels. For example, JDBC driver version.</p> <p>FINEST: The finest level. This level is used for developers only.</p> <p>ALL: This option logs all messages according to the other settings such as Trace Type and Users.</p>

Table B-7 (Cont.) Consolidator Parameters

Parameter Name	Definition
TRACE_TYPE	<p>SQL: This option logs SQL-related messages only. For example, SQL statements. Note: This option is not trace level sensitive.</p> <p>TIMING: This option logs timing data only. Note: This option is trace level sensitive. For MGP Cycle time and Synchronization time, use the Trace Level INFO option. If the MGPLogger is set to TIMING and INFO, it will log the MGP Cycle time. If the SYNCLogger is set to TIMING and INFO, it logs the synchronization time.</p> <p>DATA: This option logs data only. Note: This option is not trace level sensitive. This option prints all data with any trace level other than the OFF option.</p> <p>RESUME: Messages dealing with Reliable Transport have a RESUME trace type. This option only logs messages with Reliable Transport. Note: This option is not trace level sensitive. This option prints all the RESUME trace type messages with any trace level other than the OFF option.</p> <p>FUNCTION: This option displays the program flow by logging methods such as Entry, Exit or Invoke. For Long methods, this option logs the method's entry or exit; which is a simple invoke log. Note: This option is not trace level sensitive. This option prints all the FUNCTION trace type messages with any trace level other than the OFF option.</p> <p>GENERAL: This option logs messages that do not belong to any of the above listed trace types. Note: This type is trace level sensitive.</p> <p>ALL: This option generates logs of all trace types.</p>
TRACE_USERS	List of valid user names. The listed user trace information and system trace information is generated as output. If the value is an empty string " ", then every user is traced.
TRACE_FILE_SIZE=1	Used as the maximum file size in MB for trace files. If the value is about to be reached, the trace feature generates output to the next trace file in the pool. For more information, see TRACE_FILE_POOL_SIZE.
TRACE_FILE_POOL_SIZE=2	The default value is 2. This parameter specifies the number of files in the trace file pool. If the pool limit is reached, the trace output is overwritten to the first file in the pool. See also TRACE_FILE_POOL_SIZE.
JDBC_URL	This is the JDBC_URL used by the Sync Service and the MGP for connections to the Mobile Server Repository. If absent, it defaults to the ADMIN_JDBC_URL in the WEBTOGO section of the webtogo.ora file.

Consolidator Requirements in INIT.ORA

This document describes the Consolidator requirements for Oracle and Oracle parameter settings in the file `init.ora`. Topics include:

- [Section C.1, "Relationships Between Relevant Parameters"](#)
- [Section C.2, "Values for Processes and DML Locks"](#)

C.1 Relationships Between Relevant Parameters

You should set the following parameters in the file `init.ora` as given below:

[Table C-1](#) lists parameters that must be set in the file `init.ora`.

Table C-1 *init.ora Parameter Settings*

Parameter Name	Definition
PROCESSES	Default value: 59 to 200.
SESSIONS	Default value: Derived: $1.1 * \text{PROCESSES} + 5$
TRANSACTIONS	Default value: Derived: $(1.1 * \text{SESSIONS})$
DML_LOCKS	Default Value: Derived: $(4 * \text{TRANSACTIONS})$

C.2 Values for Processes and DML Locks

Check values for processes and `DML_LOCKS`. Massive concurrent synchronization processes use the maximum amount of resources. For each one of the concurrent clients, Consolidator requires one database connection (one session, one transaction). Therefore, the parameter value of `PROCESSES` must be set to be no less than the maximum number of concurrent clients.

During the sync, the Consolidator will make changes to the publication map tables. One DML lock is needed for each client and changed publication:

DML_LOCKS = (Number of changed publications) * (Maximum number of concurrent clients)

During the first and second sync, all publication map tables are changed for each client. So, the required DML locks are:

DML_LOCKS = (Number of publications) * (Maximum number of concurrent clients)

If you have a large number of publications, the default `DML_LOCKS` may not be sufficient. You should set it explicitly in the file **`init.ora`**. For example, CRM has approximately 50 publications. For 30 concurrent first syncs, Consolidator needs 1500 DML locks. The default value for `DML_LOCKS` with `PROCESSES` set to 200 is 1000.

Scripting Language for the Mobile Server

This document describes the scripting language for the Mobile Server. You can use scripting to perform batch processing tasks that are performed frequently by the administrator. You can write scripts for the Mobile Server in an INI text file and use the WSH tool to run your INI script. Topics include:

- [Section D.1, "Description of the Syntax"](#)
- [Section D.2, "Running a Script INI File"](#)
- [Section D.3, "Examples"](#)

D.1 Description of the Syntax

This section describes the requisite parameters and syntax to enable you to accomplish the following tasks. Topics include:

- [Creating a User](#)
- [Creating a Group](#)
- [Adding Users to a Group](#)
- [Removing Users from a Group](#)
- [Creating Access Privileges](#)
- [Granting Access](#)
- [Revoking Access](#)
- [Creating Registries](#)
- [Creating Snapshot Variables](#)
- [Deleting a User](#)
- [Deleting a Group](#)
- [Deleting Access Privileges](#)
- [Deleting a Registry](#)
- [Deleting Snapshot Variables](#)

Creating a User

Using the following syntax, you can create users to be included in a group.

```
[USER]  
NAME=<User's Name>
```

```
PASSWORD=<User's Password>
ENCRYPTED=<True or False; True if the password is encrypted, False if not>
FULLNAME=<User's Full Name>
PRIVILEGE=<User's privilege level as P, C,S, or null>
```

There are four options for setting the PRIVILEGE value for users. They are:

- **P** - Publishing an application
- **C** - Connecting to Web-to-Go
- **S** - Administering Web-to-Go
- **Null** - No privileges

Creating a Group

Using the [GROUP] script, you can create a new group (if this group does not already exist) and add listed users to the group. If you use this entry and specify the name of a group that exists, all the users in the existing group will be removed and users who are listed will be added to this group.

The following syntax enables you to create a group.

```
[GROUP]
NAME=<Group Name>
USER=<User's name you want to add to this group>
USER=<User's name you want to add to this group>
USER=<User's name you want to add to this group>
```

Adding Users to a Group

Using the [ADDUSERTOGROUP] script, you can create a new group (if this group does not already exist) and add listed users to this group. You can also use this entry to add users to an existing group.

```
[ADDUSERTOGROUP]
NAME=<Group Name>
USER=<User's name you want to add to this group>
USER=<User's name you want to add to this group>
```

Removing Users from a Group

Using the [REMOVEUSERFROMGROUP] script, you can remove listed users from a specified group.

```
NAME=<Group Name>
USER=<User's name you want to remove from this group>
USER=<User's name you want to remove from this group>
```

Creating Access Privileges

Using the [ACL] script, you can create a new ACL (if this ACL does not already exist). After creating the ACL, all the existing users will be removed and all the listed users will be added to this ACL.

Using the [GRANTACCESS] script, you can add users to the existing ACL.

The following syntax enables you to create access privileges for users and groups.

```
[ACL]
APPLICATION=<Name of the application you want to creat ACL for>
ROLE=<Role of the user; set the value as DEFAULT ROLE or ADMINISTRATIVE ROLE>
USER=<User's name>
ACCESS=<Set access status as ENABLED>
ROLE=<Role of the user>
USER=<User's name>
ACCESS=<Set access status as ENABLED>
ROLE=<Role of the group>
GROUP=<Groups's name>
ACCESS=<Set access status as ENABLED>
```

Granting Access

Using the [GRANTACCESS] script, you can create a new ACL (if this ACL does not already exist) and add listed users to this ACL.

```
[GRANTACCESS]
APPLICATION=<Name of the application you want to add ACL for>
ROLE=<Role of the user>
USER=<User's name>
ACCESS=<Access Status ENABLED/DISABLED>
ROLE=<Role of the group>
GROUP=<Groups's name>
```

Revoking Access

Using the [REVOKEACCESS] script, you can remove users that are listed in the specified ACL.

```
[REVOKEACCESS]
APPLICATION=<Name of the application you want to revoke ACL for>
ROLE=<Role of the user>
USER=<User's name>
ACCESS=<Access Status>
ROLE=<Role of the group>
GROUP=<Groups's name>
```

Creating Registries

Using the [REGISTRY] script, you can create registries.

```
[REGISTRY]
APPLICATION=<Name of the application>
NAME=<Registry Variable Name>
VALUE=<Value for this variable>
```

Creating Snapshot Variables

Using the [SNAPSHOTVAR] script, you can create snapshot variables.

```
[SNAPSHOTVAR]
```

```
NAME=<Name of the publication item>
PLATFORM=<Platform for which this publication item is>
VIRTUALPATH=<Virtual path of the application this publication item
belongs to>
USER=<Name of the user who subscribes to this application>
VAR=<Name of the Data Subsetting parameter, value of this parameter>
USER=<Name of the user who subscribes to this application>
VAR=<Name of the Data Subsetting parameter, value of this parameter>
GROUP=<Name of the group which subscribes to this application>
VAR=<Name of the Data Subsetting parameter, value of this parameter>
```

Deleting a User

Using the [DROPUSE] script, you can delete a user.

```
[DROPUSE]
NAME=<User's Name>
```

Deleting a Group

Using the [DROPGROUP] script, you can delete a group.

```
[DROPGROUP]
NAME=<Group's Name>
```

Deleting Access Privileges

Using the [DROPA] script, you can delete access privileges provided to users.

```
[DROPA]
APPLICATION=<Name of the application you want to delete ACL for>
ROLE=<Role of the user; set the value as DEFAULT ROLE or ADMINISTRATIVE ROLE>
USER=<User's name>
ACCESS=<Set access status as DISABLED>
ROLE=<Role of the group; set the value as DEFAULT ROLE or ADMINISTRATIVE ROLE>
GROUP=<Groups's name>
ACCESS=<Set access status as DISABLED>
```

Deleting a Registry

Using the [DROPREGISTRY] script, you can delete a registry.

```
[DROPREGISTRY]
APPLICATION=<Name of the application>
NAME=<Registry Variable Name>
VALUE=<Value for this variable>
```

Deleting Snapshot Variables

Using the following [DROPSNAPSHOTVAR] script, you can delete snapshot variables.

```
[DROPSNAPSHOTVAR]
NAME=<Name of the publication item>
PLATFORM=<Platform for which this publication item is>
```

```
VIRTUALPATH=<Virtual path of the application this publication item
belongs to>
USER=<Name of the user who subscribes to this application>
VAR=<Name of the Data Subsetting parameter, value of this parameter>
USER=<Name of the user who subscribes to this application>
VAR=<Name of the Data Subsetting parameter, value of this parameter>
GROUP=<Name of the group which subscribes to this application>
VAR=<Name of the Data Subsetting parameter, value of this parameter>
```

D.2 Running a Script INI File

To run a script INI file using the wsh tool, use the following command.

```
WSH -c <filename.ini> mobileadmin/manager@webtogo.world
```

D.3 Examples

This section enables you to accomplish the following tasks and describes examples from a script file in INI format. Topics include:

- [Section D.3.1, "Creating, Adding, and Granting Access"](#)
- [Section D.3.2, "Deleting, Removing, and Revoking Access"](#)

D.3.1 Creating, Adding, and Granting Access

The following examples illustrate how to create users, groups, registries, access privileges, snapshotvar template variables, add users to a group, and add users to an ACL.

```
[DATABASE]
TYPE=ORACLE

#Creation or modification of users, groups, access privileges, registry,
and snapshot variable entries using the following entries in the INI file:
#[USER], [GROUP], [ACL], [REGISTRY],[SNAPSHOTVAR].

# Create user JOHN
#
[USER]
NAME=JOHN
PASSWORD=john
ENCRYPTED=false
FULLNAME=Sample1 User John
PRIVILEGE=C

#
# Create group 'Sample Users' containing JANE, JOHN, JACK
#
[GROUP]
NAME=Sample Users
USER=JANE
USER=JOHN
USER=JACK

#
# Set the ACL on the Sample3 application.
# The following gives John, Jane, and Jack, plus all the users in the
```

```
group
# Sample Users access to the application
#
[ACL]
APPLICATION=/sample3
ROLE=Default Role
USER=JOHN
ACCESS=ENABLED
ROLE=Default Role
USER=JANE
ACCESS=ENABLED
ROLE=Default Role
USER=JACK
ACCESS=ENABLED
ROLE=Default Role
GROUP=Sample Users
ACCESS=ENABLED

#
# Add registry entry for user JOHN and a default value for the Sample3
application to the Web-to-go Repository
#
[REGISTRY]
APPLICATION=/sample3
USER=JOHN
NAME=USERCODE
VALUE=1111

#
# Add template variables.
# You can specify user/group specific values for these variables
#
[SNAPSHOTVAR]
NAME=RECORDINGS
PLATFORM=WIN32
VIRTUALPATH=/sample3
USER=JOHN
VAR=CODE, 1111
USER=JACK
VAR=CODE, 1111
USER=JANE
VAR=CODE, 2222
GROUP=Sample Users
VAR=CODE, 2222

#
#Add users to a group.
#
[ADDUSERTOGROUP]
NAME=Sample Users
USER=USER1
USER=USER2

#
#Grant Access to users.
#
[GRANTACCESS]
APPLICATION=/sample3
ROLE=Default Role
USER=USER1
```

```
ACCESS=ENABLED
ROLE=Default Role
USER=USER2
ACCESS=ENABLED
ROLE=Default Role
GROUP=Sample Users
```

D.3.2 Deleting, Removing, and Revoking Access

The following examples illustrate how to delete a user, group, registry and snapshotvar, remove users from a group, and revoke access.

```
#Deletion of users, groups, access privileges, registry and snapshot
variable entries using the following entries in
#the INI file:
#[DROPUSER], [DROPGROUP], [DROPACL], [DROPREGISTRY],[DROPSNAPSHOTVAR].

#
# Dropuser JOHN
#
[DROPUSER]
NAME=JOHN

#
# Drop group 'Sample Users'
#
[DROPGROUP]
NAME=Sample Users

#
# Drop the ACL on the sample3 application.
#
[DROPACL]
APPLICATION=/sample3
ROLE=Default Role
USER=JOHN
ACCESS=DISABLED
ROLE=Default Role
GROUP=Sample Users
ACCESS=DISABLED

#
# Drop registry entriy for user JOHN from Sample3 application.
#
[DROPREGISTRY]
APPLICATION=/sample3
USER=JOHN
NAME=USERCODE

#
# Drop template variables for user JOHN and group 'Sample Users'
#
[DROPSNAPSHOTVAR]
NAME=RECORDINGS
PLATFORM=WIN32
USER=JOHN
VAR=CODE, 1111
GROUP=Sample Users
```

```
VAR=CODE, 2222

#
#Remove users from a group.
#
[REMOVEUSERFROMGROUP]
NAME=Sample Users
USER=USER1
USER=USER2

#
#Revoke access.
#
[REVOKEACCESS]
APPLICATION=/sample3
ROLE=Default Role
USER=USER1
ACCESS=DISABLED
ROLE=Default Role
USER=USER2
ACCESS=DISABLED
ROLE=Default Role
GROUP=Sample Users
```

External Authentication

This appendix describes how to use external authentication mechanisms. Topics include:

- [Section E.1, "Using External Authentication"](#)
- [Section E.2, "External Authentication Code Sample"](#)

E.1 Using External Authentication

When a mobile user logs into the Mobile Server, the password entered by the user is compared to the user's password in the Mobile Server Repository. If the passwords match, the user password is considered authenticated and then allows access to the Mobile Server.

Instead of using the repository to store passwords, the Mobile Server verifies the user's password with an external authenticator. When a user logs in to the Mobile Server, it passes the user name and password to the external authenticator for verification. Upon successful verification, the user is allowed access to Mobile Server.

You can configure the Mobile Server to use multiple external authenticators. The Mobile Server calls the authenticators in the order specified. As soon as one of these external authenticators successfully verifies the user name and password combination, the user is considered authenticated and is allowed access to the Mobile Server. The other external authenticators are not called. When no external authenticator successfully verifies the user name/password combination, the user is denied access to Mobile Server. In order to use external authentication, you must perform the following tasks.

1. Build an external authenticator as a Java class. This class must implement the Java interface named `oracle.lite.web.spi.ExternalAuthenticator`.
2. Using the Mobile Manager, create a mobile user, but do not create a password for this user. The Mobile Server only uses external authentication for users without a specified password in the Mobile Server Repository.
3. Configure the Mobile Server to use external authentication. The names of the authenticator classes must be specified in the `[EXTERNAL_AUTHENTICATION]` section in the configuration file, **webtogo.ora**, for the Mobile Server. For example, when you add the following entries to the `[EXTERNAL_AUTHENTICATION]` section of the **webtogo.ora** configuration file, the configuration function instructs the Mobile Server to load the specified authenticator classes during startup.

```
CLASS=class1, class2, class3
```

Note: You must include such classes in the system classpath.

For example, the following code sample is an LDAP implementation, such as the Oracle Internet Directory Server, for the interface `oracle.lite.web.spi.ExternalAuthenticator`.

E.2 External Authentication Code Sample

The following code sample is an LDAP code implementation for the Oracle Internet Directory Server implementation for the interface `oracle.lite.web.spi.ExternalAuthenticator`.

Code Sample

```
import javax.naming.*;
import javax.naming.directory.*;
import com.sun.jndi.ldap.*;
import com.sun.jndi.toolkit.url.*;
import java.util.*;

public class AuthenticateJNDIUser implements
oracle.lite.web.spi.ExternalAuthenticator
{

    public AuthenticateJNDIUser()
    {
        super();
    }

    public void init()
    {
        System.out.println("calling init");
        //This method will be called when this class is initialized.
    }
    public Object authenticateUser(String uname, String pass)
    {
        try
        {
            Hashtable env = new Hashtable();
            env.put(Context.INITIAL_CONTEXT_FACTORY,
                "com.sun.jndi.ldap.LdapCtxFactory");

            String ATTRS[] = {"cn", "mail"};

            env.put(Context.SECURITY_AUTHENTICATION, "simple");
            env.put(Context.SECURITY_PRINCIPAL, "uid="+uname+", ou=People,
o=us.oracle.com");
            env.put(Context.SECURITY_CREDENTIALS, pass);
            env.put(Context.PROVIDER_URL,
"ldap://ssinghan-pc2.us.oracle.com:389");

            DirContext ctx = new InitialDirContext(env);
            System.out.println("Got InitialDirContext Successfully");
            SearchControls constraints = new SearchControls();
            constraints.setSearchScope(SearchControls.SUBTREE_SCOPE);
```

```

        NamingEnumeration results =
ctx.search("o=us.oracle.com", "uid="+uname, constraints);
        if (results ==null )
        {
            System.out.println("Null returned");
            return null;
        }

        while(results !=null && results.hasMore())
        {
            SearchResult sr = (SearchResult)results.next();
            String dn = sr.getName()+"", o=us.oracle.com";
            System.out.println("Name"+dn);
            Attributes ar = ctx.getAttributes(dn,ATTRS);

            if (ar!= null)
            {
                for(int i=0;i<=ATTRS.length-1;++i)
                {
                    System.out.println(ATTRS[i] +" : "
+ar.get(ATTRS[i]));
                }
            }
        }
        return ((Object)("Valid User"));
    }
    catch (javax.naming.NamingException ne)
    {
        System.err.println("NamingException : " + ne.getMessage());
        System.err.println("getRootCause"+ne.getRootCause());
        System.err.println("getExplanation"+ne.getExplanation());
        ne.printStackTrace();
        return null;
    }
}

public void logOff(String uname)
{ System.out.println("Into logOff : "+ uname); }

public void destroy()
{ System.out.println("Into destroy1"); }
//This method will be called when this class is unloaded.
}

```

Bypassing a Proxy Server

Users who are granted Administrator access privileges should not connect to the Mobile Server through a proxy server. The Mobile Server will not be able to see the IP addresses of their client machines and, therefore, will not be able to provide them with Administrator-level access.

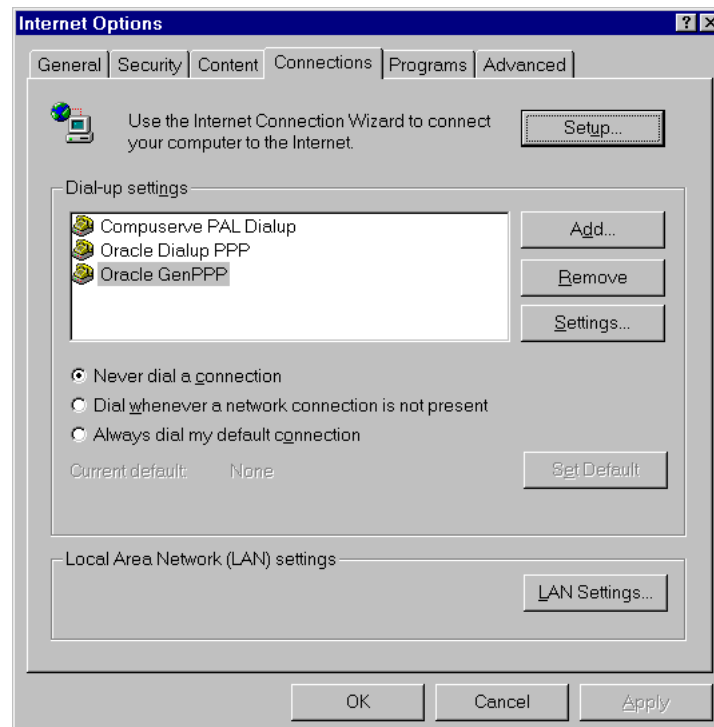
Administrators who usually connect to the network through a proxy server must bypass the proxy server before connecting to the Mobile Server.

To configure the client machine to bypass a proxy server, perform the following steps.

1. Double-click the Internet Explorer icon on your desktop and select **Internet Options** from the **Tools** menu. The browser displays the Internet Options dialog box.
2. Click the **Connections** tab.

Figure F-1 displays the Connection tab.

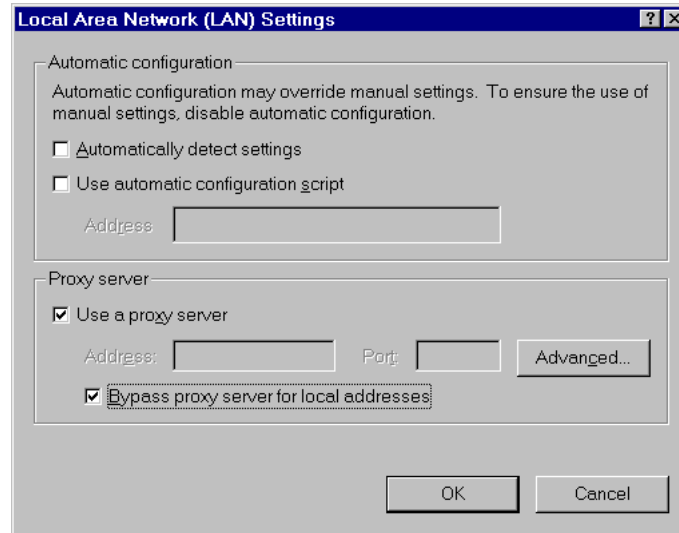
Figure F-1 Connection Tab - Internet Explorer



3. Click **LAN Settings**. Internet Explorer displays the Local Area (LAN) Network Settings dialog.

Figure F–2 displays the Local Area (LAN) Network Settings dialog.

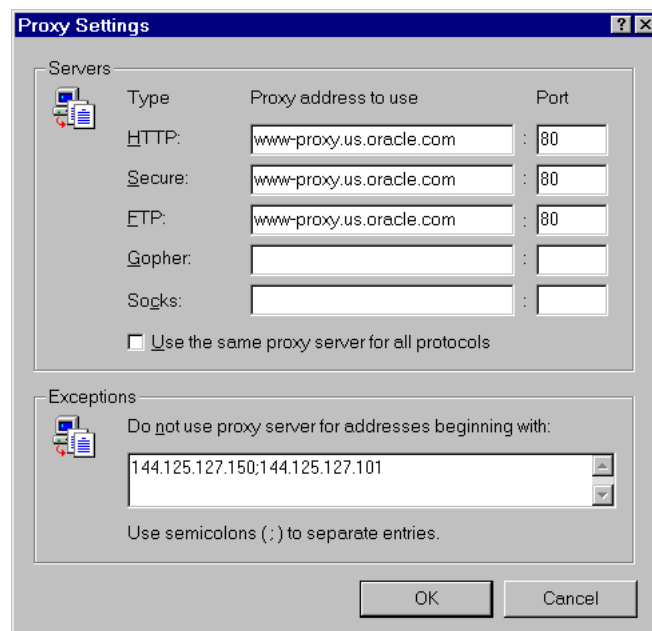
Figure F–2 Local Area Network Settings Dialog



4. Select the **Bypass Proxy Server for Local Addresses** check box.
5. Click **Advanced**. The Proxy Settings dialog appears.

Figure F–3 displays the Proxy Settings dialog.

Figure F–3 Proxy Settings Dialog



-
6. In the **Exceptions** section, in the field under **Do not use proxy server for addresses beginning with:**, enter the IP addresses of client machines for user accounts that have Administrator access.
 7. Click **OK**.

System Catalog Views

This document is a reference to system catalog views for the Mobile Admin schema. The following section lists and describes the complete set of catalog views for the Mobile Server.

G.1 Mobile Server Catalog Views

The Mobile Admin schema is installed as part of the Mobile Server during installation. However, the Mobile Admin schema is not part of the Mobile Development Kit.

The system catalog views are read-only and should not be modified. The views include:

- [Section G-1, "ALL_CLIENTS Parameters"](#)
- [Section G-2, "ALL_ERROR Parameters"](#)
- [Section G-3, "ALL_PUBLICATIONS Parameters"](#)
- [Section G-4, "ALL_SUBSCRIPTIONS Parameters"](#)
- [Section G-5, "ALL_SEQUENCES Parameters"](#)
- [Section G-6, "ALL_SEQUENCE_PARTITIONS Parameters"](#)
- [Section G-7, "ALL_PUBLICATION_ITEMS_ADDED Parameters"](#)
- [Section G-8, "ALL_PUBLICATION_ITEMS Parameters"](#)
- [Section G-9, "ALL_PUBLICATION_ITEM_INDEXES Parameters"](#)
- [Section G-10, "ALL_SUBSCRIPTION_PARAMS Parameters"](#)

G.1.1 "MOBILEADMIN".CV\$ALL_CLIENTS

The "MOBILEADMIN".CV\$ALL_CLIENTS view provides information about Mobile Server clients.

[Table G-1](#) provides a description of ALL_CLIENT parameters.

Table G-1 ALL_CLIENTS Parameters

Column	Datatype	Null	Description
CLIENT	VARCHAR (30)	NULL	The Mobile Server client
LASTREFRESH_STARTTIME	VARCHAR (19)	NULL	Start time of the last refresh session
LASTREFRESH_ENDTIME	VARCHAR (19)	NULL	End time of the last refresh session

G.1.2 "MOBILEADMIN".CV\$ALL_ERROR

The "MOBILEADMIN".CV\$ALL_ERROR view provides information about failed client transactions.

[Table G–2](#) provides a description of ALL_ERROR parameters.

Table G–2 ALL_ERROR Parameters

Column	Datatype	Null	Description
CLIENT	VARCHAR(30)	NOT NULL	Client to which the failed transaction belongs.
TRANSACTION_ID	NUMBER(10)	NOT NULL	ID of the failed transaction.
ITEM_NAME	VARCHAR2(30)	NOT NULL	Name of the publication item that failed.
MESSAGE_TEXT	VARCHAR2(2048)	NOT NULL	Error text associated with the failed transaction and publication item.

G.1.3 "MOBILEADMIN".CV\$ALL_PUBLICATIONS

The ALL_PUBLICATIONS view provides information about Mobile Server publications.

[Table G–3](#) provides a description of ALL_PUBLICATIONS parameters.

Table G–3 ALL_PUBLICATIONS Parameters

Column	Datatype	Null	Description
NAME	VARCHAR2(30)	NULL	Publication Name.
TYPE	VARCHAR2(40)	NULL	Publication Type.
NAME_TEMPLATE	VARCHAR2(30)	NULL	Snapshot Name Template.
ENFORCE_RI	CHAR(1)	NOT NULL	Reserved.

G.1.4 "MOBILEADMIN".CV\$ALL_SUBSCRIPTIONS

The ALL_SUBSCRIPTIONS view provides information about Mobile Server subscriptions.

[Table G–4](#) provides a description of ALL_SUBSCRIPTION parameters.

Table G–4 ALL_SUBSCRIPTIONS Parameters

Column	Datatype	Null	Description
CLIENT	VARCHAR2(30)	NULL	The subscription's clients.
PUBLICATION	VARCHAR2(30)	NULL	The subscription's publication.
INSTANTIATED	CHAR(1)	NULL	A boolean value that indicates whether the subscription is instantiated.

G.1.5 "MOBILEADMIN".CV\$ALL_SEQUENCES

The ALL_SEQUENCES view provides information about Mobile Server sequences.

[Table G–5](#) provides a description of ALL_SEQUENCES parameters.

Table G–5 ALL_SEQUENCES Parameters

Column	Datatype	Null	Description
NAME	VARCHAR2 (30)	NULL	The sequence name.

G.1.6 "MOBILEADMIN".CV\$ALL_SEQUENCE_PARTITIONS

The ALL_SEQUENCE_PARTITIONS view provides information about Mobile Server sequence partitions.

[Table G–6](#) provides a description of ALL_SEQUENCE_PARTITIONS parameters.

Table G–6 ALL_SEQUENCE_PARTITIONS Parameters

Column	Datatype	Null	Description
CLIENT	VARCHAR2 (30)	NULL	The client to which the sequence is assigned.
NAME	VARCHAR2 (30)	NULL	The sequence name.
CURR_VALUE	NUMBER (38)	NULL	The current sequence value.
INCREMENT_BY	NUMBER (38)	NULL	The sequence's increment value. The sequence increments based on this number.

G.1.7 "MOBILEADMIN".CV\$ALL_PUBLICATION_ITEMS_ADDED

The ALL_PUBLICATION_ITEMS_ADDED view provides information about Mobile Server publication items.

[Table G–7](#) provides a description of ALL_PUBLICATION_ITEMS_ADDED parameters.

Table G–7 ALL_PUBLICATION_ITEMS_ADDED Parameters

Column	Datatype	Null	Description
PUB_NAME	VARCHAR2 (30)	NULL	The publication name.
ITEM_NAME	VARCHAR2 (30)	NULL	The publication item name.
OWNER	VARCHAR2 (30)	NOT NULL	The base object owner.
OBJECT_NAME	VARCHAR2 (30)	NOT NULL	The base object name.
TEXT	VARCHAR2 (2048)	NOT NULL	The select statement.
UPDATABLE	VARCHAR2 (1)	NULL	The updatable option.
REFRESH_METHOD	CHAR (1)	NOT NULL	The refresh method. Options include fast refresh and complete refresh.
WINNING_RULE	VARCHAR2 (30)	NULL	The winning rules option for resolving replication conflicts. Options include "client wins" and "server wins".

G.1.8 "MOBILEADMIN".CV\$ALL_PUBLICATION_ITEMS

The ALL_PUBLICATION_ITEMS view provides information about Mobile Server publication items.

[Table G–8](#) provides a description of ALL_PUBLICATION_ITEMS parameters.

Table G–8 *ALL_PUBLICATION_ITEMS Parameters*

Column	Datatype	Type	Description
NAME	VARCHAR2 (30)	NULL	The publication item name.
OWNER	VARCHAR2 (30)	NOT NULL	The owner of the publication items' base object.
OBJECT_NAME	VARCHAR2 (30)	NOT NULL	Name of the base object.
TEXT	VARCHAR2 (2048)	NOT NULL	The select statement.
REFRESH_METHOD	CHAR (1)	NOT NULL	The refresh method. Options include fast refresh and complete refresh.

G.1.9 "MOBILEADMIN.CV\$ALL_PUBLICATION_ITEM_INDEXES

The ALL_PUBLICATION_ITEM_INDEXES view provides information about Mobile Server publication item indexes.

[Table G–9](#) provides a description of ALL_PUBLICATION_ITEM_INDEXES parameters.

Table G–9 *ALL_PUBLICATION_ITEM_INDEXES Parameters*

Column	Datatype	Null	Description
NAME	VARCHAR2 (30)	NULL	Index name.
PUB_ITEM	VARCHAR2 (30)	NOT NULL	Publication item name.
INDX_TYPE	CHAR (1)	NOT NULL	Index type.
COLUMN_LIST	VARCHAR2 (2048)	NOT NULL	Column list.

G.1.10 "MOBILEADMIN"CV.\$ALL_SUBSCRIPTION_PARAMS

The ALL_SUBSCRIPTION_PARAMS view provides information about Mobile Server subscription parameters.

[Table G–10](#) provides a description of ALL_SUBSCRIPTION_PARAMS parameters.

Table G–10 *ALL_SUBSCRIPTION_PARAMS Parameters*

Column	Datatype	Null	Description
NAME	VARCHAR2 (30)	NULL	Publication name.
CLIENT	VARCHAR2 (30)	NULL	Client name.
PARAM_NAME	VARCHAR2 (30)	NULL	Parameter name.
PARAM_VALUE	VARCHAR2 (30)	NULL	Parameter value.

Using the JDBC Thin Driver

To use the JDBC thin driver with Mobile Server applets, you will need to accomplish the following tasks.

1. Upload the Oracle JDBC driver file **Classes12.zip** to the Mobile Server Repository.
2. Change the applet tag of the html page to include **Classes12.zip** in the **Archive** tag.

The following sections describe these steps in detail. Topics include:

- [Section H.1, "Upload the Oracle JDBC Driver"](#)
- [Section H.2, "Change the Applet Tag"](#)

H.1 Upload the Oracle JDBC Driver

To upload the Oracle JDBC driver file **Classes12.zip** to the Mobile Server Repository, perform the following steps.

1. Start the WSH tool and enter the following statement to connect to the Mobile Server Repository.

```
WSH -o mobileadmin/manager@webtogo.world
```

Note: In the above statement `manager` represents the password.

2. Start the Command Prompt and enter the following.

```
WEBTOGO.WORLD:/> cd webtogo
```

```
WEBTOGO.WORLD:/webtogo>copy %ORACLE_HOME%\  
jdbc\lib\classes12.zip
```

```
WEBTOGO.WORLD:/webtogo>exit
```

H.2 Change the Applet Tag

To change the applet tag, perform the following steps.

1. Change the applet tag of the HTML page to include **Classes12.zip** in the **ARCHIVE** tag.
2. Add the following **ARCHIVE** tag to all the HTML pages containing the `<APPLET>` tag.

```
ARCHIVE="/webtogo/classes12.zip"
```

3. If the **ARCHIVE** tag already exists, you must modify it as follows.

```
ARCHIVE="/webtogo/classes12.zip, testapp.jar"
```

POLITE.INI Database Parameters

You can customize your Oracle Database Lite by changing the database parameter values defined in your `POLITE.INI` file. This document discusses the `POLITE.INI` file and its associated parameters. Topics include:

- [Section I.1, "POLITE.INI File Overview"](#)
- [Section I.2, "POLITE.INI Parameters"](#)
- [Section I.3, "Sample POLITE.INI File"](#)

I.1 POLITE.INI File Overview

The `POLITE.INI` file centralizes database volume ID assignments and defines parameters for all databases on a system. When you install Oracle Database Lite, the installation creates the `POLITE.INI` file in your Windows 98, NT, 2000, or XP home directory.

The installation automatically sets the parameters in your `POLITE.INI` file, but you can modify them to customize the product's behavior. To modify the `POLITE.INI` file, you can use an ASCII text editor.

I.2 POLITE.INI Parameters

The following sections describe the parameters in the All Databases section of the `POLITE.INI` file.

I.2.1 CacheSize

Specifies the size of the object cache in kilobytes. The minimum is 128. If not set, the default is 4096 (4 megabytes).

I.2.2 Database ID

Defines the next Database Volume ID number to be assigned the `CREATE DATABASE SQL` command. Database Volume ID numbers must be unique for each database file on the system.

I.2.3 DBCharEncoding

Specifies the Oracle Database Lite character set. If set to `NATIVE`, the default is the system default character set.

[Table I-1](#) lists the supported code pages and their corresponding values of `DbCharEncoding`.

Table I-1 Supported Code Pages and Values

Code Page	DbCharEncoding	Language
(1250)	ee8mswin1250	(Croatian, Czech, Hungarian, Polish, Romanian, Slovak, and Slovenian)
(1251)	c18mswin1251	(Bulgarian, Russian, and Ukrainian)
(1252)	we8mswin1252	(English (United States), Catalan, Danish, Dutch (Netherlands), English (United Kingdom), Finish, French (France), German (Germany), Icelandic, Italian (Italy), Malay (Malaysia), Norwegian (Bokmal), Portuguese (Brazil), Portuguese (Portugal), Spanish (Mexico), Spanish (Spain), and Swedish)
(1253)	el8mswin1253	(Greek)
(1254)	tr8mswin1254	(Turkish)
(1255)	iw8mswin1255	(Hebrew)
(1256)	ar8mswin1256	(Arabic (Egypt), and Arabic (UAE))
(1257)	blt8mswin1257	(Estonian and Lithuanian)
(932)	ja16sjis	(Japanese)
(936)	zhs16gbk	(Chinese (PRC) and Chinese (Singapore))
(949)	ko16mswin949	(Korean)
(950)	zht16mswin950	(Chinese (Taiwan) and Chinese (Hong Kong))

I.2.4 MAXINDEXCOLUMNS

Defines the number of columns used in the index creation statement. For more information, see Index Creation Options, in the *Oracle Database Lite SQL Reference*.

I.2.5 NLS_DATE_FORMAT

Allows you to use a date format other than the Oracle Database Lite default. When a literal character string appears where a date value is expected, the Oracle Database Lite tests the string to see if it matches the formats of Oracle, SQL-92, or the value specified for this parameter in the POLITE.INI file. Setting this parameter also defines the default format used in the TO_CHAR or TO_DATE functions when no other format string is supplied.

For Oracle, the default is dd-mon-yy or dd-mon-yyyy. For SQL-92, the default is yy-mm-dd or yyyy-mm-dd.

Using RR in the format forces two digit years less than or equal to 49 to be interpreted as years in the 21st century (2000–2049), and years 50 and over, as years in the 20th century (1950–1999). Setting the RR format as the default for all two digit year entries allows you to become year-2000 compliant. For example,

```
NLS_DATE_FORMAT= 'RR-MM-DD'
```

You can also modify the date format using the ALTER SESSION command. For more information, see the *Oracle Database Lite SQL Reference*.

I.2.5.1 Date Format

A date format includes one or more of the elements listed in the following table. Elements that represent similar information cannot be combined, for example, you

cannot use SYYYY and BC in the same format string. [Table I-2](#) lists date formats and their corresponding description.

Table I-2 Date Formats

Format	Description
AM or P . M .	Meridian indicator, periods are optional.
PM or P . M .	Meridian indicator, periods are optional.
CC or SCC	Century, "S" prefixes BC dates with "-".
D	Day of week.
DAY	Name of day, padded with blanks to length of 9 characters.
DD	Day of month (1-31).
DDD	Day of year (1-366).
DY	Abbreviate name of day.
IW	Week of year (1-52 or 1-53) based on the ISO standard.
IYY, IY, or I	Last 3, 2, or 1 digit(s) of the ISO year, respectively.
IYYY	4-digit year, based on the ISO standard.
HH or HH12	Hour of the day (1-12).
HH24	Hour of the day (0-23).
MI	Minute (0-59).
MM	Month (01-12, for example, JAN=01).
MONTH	Name of the month, padded with blanks to length of 9 characters.
MON	Abbreviated name of the month.
Q	Quarter of the year, (1,2,3,4, for example, JAN-MAR=1).
RR	Last 2 digits of the year, for years in other countries. This forces two-digit years less than or equal to 49 to be interpreted as years in the 21st century (2000-2049), and years 50 and over, as years in the 20th century (1950-1959).
WW	Week of the year (1-53), where 1 starts on the first day of the year and continues to the seventh day of the year.
SS	Second (0-59).
SSSSS	Seconds past midnight (0-86399).
Y or YYY	Year with comma in this position.
YEAR or SYEAR	Year, spelled out. "S" prefixes BC dates with "-".
YYYY or SYYYY	4-digit year. "S" prefixes BC dates with "-".
YYY, YY, or Y	Last 3, 2, or 1 digit(s) of the year.

I.2.5.2 Date Format Examples

Listed below are sample variations of the NLS_DATE_FORMAT parameter.

1. YYYYY-MONTH-DAY : HH24 : MI : P . M .
2. YYYYY/MONTH/DD , HH24 : MI A . M .
3. YYYYY-MONTH-DAY : HH24 : MI : PM

4. MM D, YYYY, HH:MI A.M.
5. MM, WW, RR, HH:MI A.M.
6. MM, IW, RR, HH:M1 A.M.
7. MM, DY, RR, HH:MI A.M.
8. MM; DY; IYY, HH:MI A.M.
9. MON WW, RR, HH:MI A.M.
10. MONTH.DD, SYYYY, HH:MI A.M.
11. MONTH/DD, YYYY, HH:MI A.M.
12. MONTH|DD, YYYY, HH:MI A.M.
13. MONTH DD, YYYY, HH:SSSS:MI A.M.
14. MONTH DD, HH:SS::MI CC
15. MONTH DD, HH:SS:MI SCC
16. MONTH W, YYYY, HH:MI A.M.
17. MONTH WW, YYYY, HH:MI A.M.
18. MONTH WW, RR, HH:MI A.M.
19. MONTH WW, Q, HH:MI A.M.
20. MONTH WW, RR, HH:MI A.M.

I.2.6 NLS_LOCALE

Defines the NLS_LOCALE parameter in the POLITE.INI file to specify the locale data of Oracle Database Lite. Oracle Database Lite locale data includes the following items.

- Decimal character and group separator
- Locale currency symbol and ISO currency symbol
- Day, week, month names, and their abbreviations

For example, NLS_LOCALE=FRENCH_FRANCE specifies the locale data of french_french in Oracle Database Lite. [Table I-3](#) describes the supported locale and corresponding values of the NLS_LOCALE setting.

Table I-3 Supported Locales and Values

Locale	NLS_LOCALE
English (United States)	AMERICAN_AMERICA
Arabic (Egypt)	ARABIC_EGYPT
Arabic (UAE)	ARABIC_UNITED ARAB EMIRATES
Bulgarian	BULGARIAN_BULGARIA
Catalan	CATALAN_CATALONIA
Chinese (PRC)	SIMPLIFIED CHINESE_CHINA
Chinese (Singapore)	SIMPLIFIED CHINESE_SINGAPORE
Chinese (Taiwan)	TRADITIONAL CHINESE_TAIWAN
Chinese (Hong Kong)	TRADITIONAL CHINESE_HONG KONG
Croatian	CROATIAN_CROATIA

Table I-3 (Cont.) Supported Locales and Values

Locale	NLS_LOCALE
Czech	CZECH_CZECH REPUBLIC
Danish	DANISH_DENMARK
Dutch (Netherlands)	DUTCH_THE NETHERLANDS
English (United Kingdom)	ENGLISH_UNITED
Estonian	ESTONIAN_ESTONIA
Finnish	FINNISH_FINLAND
French (France)	FRENCH_FRANCE
German (Germany)	GERMAN_GERMANY
Greek	GREEK_GREECE
Hebrew	HEBREW_ISRAEL
Hungarian	HUNGARIAN_HUNGARY
Icelandic	ICELANDIC_ICELAND
Italian (Italy)	ITALIAN_ITALY
Japanese	JAPANESE_JAPAN
Korean	KOREAN_KOREA
Lithuanian	LITHUANIAN_LITHUANIA
Malay (Malaysia)	MALAY_MALAYSIA
Norwegian (Bokmal)	NORWEGIAN_NORWAY
Polish	POLISH_POLAND
Portuguese (Brazil)	BRAZILIAN PORTUGUESE_BRAZIL
Portuguese (Portugal)	PORTUGUESE_PORTUGAL
Romanian	ROMANIAN_ROMANIA
Russian	RUSSIAN_CIS
Slovak	SLOVAK_SLOVAKIA
Slovenian	SLOVENIAN_SLOVENIA
Spanish (Mexico)	MEXICAN SPANISH_MEXICO
Spanish (Spain)	SPANISH_SPAIN
Swedish	SWEDISH_SWEDEN
Turkish	TURKISH_TURKEY
Ukrainian	UKRANIAN_UKRAINE

I.2.7 NLS_SORT

This parameter can be used to define the collation sequence for databases created on the Oracle Database Lite instance. Collation is referred as ordering strings into a culturally acceptable sequence. A collation sequence is a sequence of all collation elements from an alphabet from the smallest collation order to the largest.

NLS_SORT=[collation sequence]

When this parameter is used, all databases created with the `CREATEDB` command line utility or those that are replicated from the Mobile Server are enabled for the collation sequence unless a different collation sequence is specified when using the utility. Languages currently supported are `BINARY` (default), `FRENCH`, `GERMAN`, `CZECH`, and `XCZECH`.

Note: Unless you require your databases to have linguistic sort enabled for a supported collation sequence, it is recommended that you use the `CREATEDB` utility with the `NLS_SORT <collation sequence>` parameter, which overrides this `POLITE.INI` parameter. Setting the `NLS_SORT` using the `POLITE.INI` file means that your databases have the specified collation sequence enabled. There is currently no way to convert a database from one collation sequence to another.

For a complete description of this feature, see Section 3.3, "CREATEDB," in the *Oracle Database Lite Tools and Utilities Guide*.

I.2.8 ReverseJoinOrder

Specifies the order of tables joined for a query. Options are `TRUE` or `FALSE`. If set to `TRUE`, the tables are joined in the reverse order in which they are given in the `FROM` clause. If `FALSE`, the tables are joined in the same order as they are given in the `FROM` clause. If not set, the Oracle Database Lite query optimizer determines the optimal join order. This option is recommended for advanced use as it disables the `JOIN` optimization for all queries. To optimize a single query, it is suggested that you use `HINTS` instead. For more information, see the Oracle Database Lite SQL Reference.

I.2.9 SharedAddress

To manage the data needed across applications, Oracle Database Lite uses shared memory. Oracle Database Lite attaches the shared memory at a specific location of the process memory. In very few cases, this location may already be in use by other tools resulting in an error. To address this issue, Oracle Database Lite supports the following protocol to determine the memory address to attach for shared memory.

Before attaching shared memory, Oracle Database Lite examines the `SharedAddress` and `SuggestedSharedAddress` variables that specify a hexadecimal 32-bit address (for example, 18000000). Oracle Database Lite uses the first value it finds. If you do not set either variable, Oracle Database Lite first tries the address 30000000. This value is above the range used by most applications.

If an Oracle Database Lite client is already running, and the second process cannot get the same shared memory address, it fails with an error. However, it also finds an address that is available in the second process and writes it to the **POLITE.ini** file as the `SuggestedSharedAddress`. If the user exits all Oracle Database Lite clients and runs the same mix of applications, the problem does not reoccur.

If automatic conflict resolution fails, you should modify the `SharedAddress` variable until you resolve the issue. For example, you can try values spaced by 256 MB: 20000000, 24000000, 28000000, and so on.

I.2.10 SuggestedSharedAddress

For a detailed description, see [SharedAddress](#).

I.2.11 SQL Compatibility

Oracle Database Lite supports both Oracle SQL and SQL-92 features. For more information on Oracle SQL and SQL-92, see the *Oracle Database Lite SQL Reference*.

If there is a conflict between Oracle SQL and SQL-92, the `SQLCompatibility` flag is referenced. If you specify `ORACLE` for the parameter, Oracle SQL is favored, and if you specify `SQL92`, SQL-92 is favored. If you do not include this parameter in the **POLITE.ini**, Oracle SQL is favored, by default.

I.2.12 TempDB

The temporary database is created by default in the main memory. This improves the performance of some queries that require the use of temporary tables. Unless you explicitly choose to create the temporary database in the file system, the `poltempx.odb` files are not created. The `*.slx` files that are sometimes used to store savepoint information are also not created. If you plan to create a large result set, you must either have enough swap space to hold the result, or choose the file option for the temporary database.

To include this option, use the following syntax in the **POLITE.INI** file.

```
TempDB=<path temporary_database_name>
```

For example,

```
TempDB=c:\temp\olite_
```

As a result of the example setting, Oracle Database Lite creates temporary databases as given below.

```
c:\temp\olite_0.odb, c:\temp\olite_1.odb, ...
```

I.2.13 TempDir

Specifies the directory where the temporary database **POLTEMP.ODB** is created. If not set, the default is any `TEMP`, `TMP` or `WINDIR` setting defined in your environment.

I.2.14 OLITE_SQL_TRACE

Generates the SQL statement text, compilation time, execution plan, and the bind value.

For example,

```
OLITE_SQL_TRACE= yes
```

SQL trace output is dumped to a trace file named `olddb_trc.txt` in the current working directory of the database process. For a database service on Windows, Windows NT or the Oracle Database Lite daemon for a Linux platform, the current working directory is specified by the `wdir` parameter during the database startup service or daemon. Applications that use an embedded connection to connect to the database contain a working directory. This working directory is the application's working directory. To implement the tracing feature, the database process must contain permissions to create the trace file in the current working directory. The trace output is always included in the trace file. If the trace file does not exist, it is created automatically.

I.2.15 FLUSH_AFTER_WRITE

Syntax

FLUSH_AFTER_WRITE={YES|TRUE|FALSE|NO}

Default Value

FALSE

By default, the parameter `FLUSH_AFTER_WRITE` is disabled. Hence, writes to a database are not flushed. The last write operation during a `COMMIT` operation always flushes file buffers, thereby eliminating the danger of losing data. For devices that are unreliable, users can enable this flag and set the parameter to `TRUE` or `YES`. When enabled, every write action flushes file buffers. However, this setting degrades the database `COMMIT` performance.

Note: This parameter applies to the WinCE platform only.

I.3 Sample POLITE.INI File

The following content is displayed from a sample `POLITE.INI` file.

```
[All Databases]
DatabaseID=128
DBCharEncoding=NATIVE
SuggestedSharedAddress=10270000
CacheSize=4096
MAXINDEXCOLUMNS=5
SQLCompatibility=SQL92
NLS_Date_Format=RR/MM/DD H24,MI,SS
NLS_Locale=ENGLISH
TempDB=c:\temp\olite_
TempDir=D:\TMP
```

On the WinCE platform, include the parameter `DATADIRECTORY=\Storage Card:\Orace:\tmp`. To synchronize, run `mSync.exe`.

The Consolidator Client API (OCAPI)

This document provides a reference to the Consolidator Client API parameters, enabling you, the Administrator, to modify the settings of each parameter. Topics include:

- [Section J.1, "Overview"](#)
- [Section J.2, "Sync Client Parameters"](#)

J.1 Overview

The Consolidator Client API (OCAPI) is a set of functions that allows programs on client devices to set synchronization parameters and start a synchronization session. You can also use this API to monitor the progress of the synchronization session. OCAPI is the interface to the client side synchronization engine.

As the Administrator, you can set the OCAPI parameters to change the default behavior of OCAPI. The parameters are in the file **polite.ini** under the [SYNC] section on Windows 32. On Windows CE and EPOC, the file name is **polite.txt**. This feature is not supported on Palm.

When you set the OCAPI parameters in the file **polite.ini** or in the file **polite.txt** (or in both files), the parameter settings will be implemented for the client, based on the client platforms where the parameter settings need to apply.

An OCAPI function communicates with the Mobile Server through the selected transport means and synchronizes the local database with the remote Mobile Server. When the client application starts a synchronization session, the OCAPI parameters will be implemented.

J.2 Sync Client Parameters

The following sections list the OCAPI parameters with their corresponding description and example. OCAPI provides you with the following support functions:

- Enable the caller to start the synchronization process from the client side.
- Set flags for the synchronization session.
- Save user information locally.

Note: OCAPI is only supported on the Windows 32, Windows CE, and EPOC platforms. On the Palm platform, the developer must set all options in the structure `ocEnv`. For more information, see the *Oracle Database Lite Developer's Guide*.

J.2.1 TIME_LOG

This parameter enables the Administrator to record the start and end time of a synchronization operation. If this parameter is set, OCAPI creates a table called C\$SYNC_TIME in the conscli.odb file. This file logs the duration of every synchronization process. OCAPI inserts a record in the C\$SYNC_TIME table which stores the start and end time of every synchronization operation. Setting this parameter enables the Administrator to maintain a log history of synchronization times.

Example

TIME_LOG=1

The above value creates a table called C\$SYNC_TIME and inserts one row containing the start and end time of the synchronization process.

Default Value

0

The above value indicates that the timelog feature is off.

J.2.2 UPDATE_LOG

This parameter enables the Administrator to set the update log file. If this parameter is set, OCAPI creates a table called C\$UPDATE_LOG in the conscli.odb file. For every DML operation received from the server, OCAPI records each operation in the C\$UPDATE_LOG table. Each record contains three entries namely Table Name, Client Side Row ID, and the Log Action Type. The Table Name refers to the table that the operation is performed on. The Client Side Row ID (C\$UID) is a record pointer that points to the record's Row ID. Type refers to the type of DML operation such as update, insert, and delete.

Example

UPDATE_LOG=1

The above value creates and inserts rows in the C\$UPDATE_LOG file.

Default Value

0

J.2.3 COMPRESSION

This parameter enables the administrator to set compression limits for data transmissions. The client dictates if the transmitted data is compressed or not. If the data transmitted by the client to the server is compressed, the client receives compressed data from the server.

Example

COMPRESSION=1

This parameter invokes the compression feature.

Default Value

1

The above value indicates that the compression feature is on.

J.2.4 DEBUG

This parameter enables the Administrator to view debugging messages that are sent to the file `debug.txt`. This file includes the database name, table names, and the DML operation. When this parameter is set to 1, the debug information regarding the database name, table names, and the DML operation goes into the file **debug.txt**. This enables OCAPI to invoke debugging messages.

Default Value

0

J.2.5 AUTO_COMMIT_COUNT

This parameter enables the Administrator to invoke the automatic commit count feature. If this parameter is set to 0, OCAPI calls a commit count at the end of processing for each publication. If this parameter is set to 1000, OCAPI calls commits for every 1000 inserts. This parameter must be used only during the complete refresh process of the system.

Default Value

0 for Win32

250 for WinCE

J.2.6 TEMP_DIR

This parameter enables the Administrator to set a directory for temporary file creation. OCAPI creates a temporary file for saving retrieved data. The Administrator can also specify the directory for the temporary file. When a large volume of data is being synchronized, the data received in the temporary file can be written to a flash card to save the system's memory. This feature is beneficial for WinCE developers.

Example

```
TEMP_DIR=\Storage Card
```

OCAPI creates a temporary file on the storage card of the Windows CE application. It saves the main memory allocated for the application.

J.2.7 RESUME_CLIENT_TIMEOUT

This parameter enables the Administrator to set the total number of seconds that the client should use to resume network timeout operations.

Default Value

60 seconds

Example

```
RESUME_CLIENT_TIMEOUT=120
```

J.2.8 RESUME_CLIENT_MAXSEND

This parameter enables the Administrator to set the maximum data size in kilo bytes sent by a client in a single POST request. Some proxies maintain fixed limits on data size in one request.

Default Value

1MB (1024KB)

Example

RESUME_CLIENT_MAXSEND=2048

J.2.9 ERROR_REPORT

This parameter enables the Administrator to set client synchronization report results for the server.

If set to 0, reports errors to the server during the next synchronization process.

If set to 1, reports errors and creates an extra connection to the server.

If set to 2, reports synchronization success or error cases and creates an extra connection to the server.

Default Value

0

Example

ERROR_REPORT=2

J.2.10 DB_ENCODING

This parameter enables the Administrator to specify client DB character encoding. This parameter value is the same as values used in Java character encoding. For more information about Java encoding, refer the following URL.

<http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.html>

This character encoding affects CHAR and VARCHAR datatypes inside client snapshot tables only.

Default Value

NULL

The default value indicates a native character set.

J.2.11 MEM_THRESHOLD

This parameter enables the Administrator to set the memory threshold value in bytes during synchronization. OCAPI stops synchronization operations when the available memory is less than the specified value. Under low memory conditions, applications can be unstable on a Windows CE device. By using this parameter's settings, OCAPI prevents low memory conditions. If the available memory is lower than this value, OCAPI displays an error message.

Default Value

524288 (512 KB)

J.2.12 VALIDATEDB

This parameter enables the Administrator to validate the Oracle Lite database, using the **validatedb.exe** after the synchronization process. When an error is reported by the

validatedb.exe, OCAPI reports the error to the server. You can set this parameter value from 0 to 100. If set to 100, OCAPI runs the validatedb.exe for every synchronization process. If set to 50, OCAPI runs the validatedb.exe for every alternate synchronization process. If set to 1, OCAPI runs the validatedb.exe, once for every 100 synchronization processes.

Default Value

0

J.2.13 ENCRYPTDB

This parameter enables the Administrator to encrypt the Oracle database, using the **encryptdb.exe** after synchronization. If set to 2, the encryptdb.exe runs after every synchronization process. If set to 1, the encryptdb.exe runs only when a new Oracle Lite database (ODB) file is created.

Default Value

0

Glossary

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 Mobile Server 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. See also "Primary Key".

Index

An index is a database object that provides fast access to individual rows in a table. You create an index to accelerate queries and sorting operations performed against the table's data. Indexes can also be used 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.

JavaServer Pages

JavaServer Pages (JSP) is a technology that enables developers to change a page's layout without altering the page's underlying content. JSP 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 Server 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 an 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 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 mobile application that runs in the browser for easy administration of applications and users. Administrators use the Mobile Manager to perform such functions as granting or revoking application access to users or groups, modifying snapshot template variables, or deleting applications from the Mobile Server.

Mobile Server

The Mobile Server resides on the application server tier of the three-tier Mobile Server model and processes requests from Mobile Clients to modify data in the database server.

Mobile Server Repository

The Mobile Server Repository is a virtual file system that resides on Oracle. 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 Lite database. To view this information, you can use SQL*Plus.

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 Lite. See also "Online Mode".

Oracle

Oracle 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 Oracle.

Oracle Lite

Oracle 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 Lite.

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 developers to define and package new or existing Mobile Server applications.

Positioned Delete

A positioned `DELETE` statement deletes the current row of the cursor. Its format is as follows:

```
DELETE FROM table
      WHERE CURRENT OF cursor_name
```

Positioned Update

A positioned `UPDATE` statement updates the current row of the cursor. Its format is as follows:

```
UPDATE table SET set_list
      WHERE CURRENT OF cursor_name
```

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.

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 a 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. Replication conflicts can be avoided by proper subsetting of data. The Packaging Wizard allows the developer to specify rules on how to handle conflicts.

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.

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 real-time from the Oracle database and downloads the same 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.

SQL*Plus

SQL*Plus is a tool that connects to an Oracle Lite database and accesses data. You must have ODBC support enabled to use SQL*Plus.

Switching Modes

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

Synchronization

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

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. See also "SQL".

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 Oracle. It is centrally managed from the server and Web-to-Go applications can be run when Web-to-Go is connected to the server (online) or disconnected from the server (offline). When Web-to-Go is offline, it stores data locally in the Cache folder and synchronizes data with the server, when it goes back online.

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 Server and the Oracle Lite database. Web-to-Go replicates the user's applications and data to Oracle Lite when the user switches to offline mode. When the user switches back to online mode, Web-to-Go replicates any data changes to Oracle.

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

A

- Adding Bookmarks, 4-8
- Adding New Groups, 5-3
- Adding New Users, 5-4
- Administering Mobile Applications
 - Adding Web Application Archive (WAR) Files, 4-7
 - Creating Database Tables, 4-5
 - Listing Applications, 4-1
 - Modifying Data Subsetting Parameters, 4-5
 - Overview, 4-1
 - Removing Applications from the Mobile Server, 4-5
 - Uploading Applications to the Mobile Server Repository, 4-4
- Administering the Mobile Server
 - Resuming the Server, 7-2
 - Suspending the Server, 7-2
 - Viewing System Status Reports for the Server, 7-1
- Administering Users
 - Overview, 5-1
 - Standalone Mode, 5-1

B

- Branch Office
 - Administration, 15-10
 - Architecture, 15-7
 - Connecting Clients to the Branch Office Database Machine, 15-9
 - The Branch Office Environment, 15-7
 - Branch Office Installation and Configuration, 15-3
 - Introduction, 15-1
 - Terms and Concepts, 15-3

C

- certificate authority
 - examples, 17-2
- Checking Sync Service Alerts, 8-4
- Consolidator Requirements in INIT.ORA
 - Relationships Between Relevant Parameters, C-1
 - Values for Processes and DML Locks, C-1

D

- Deployment
 - Creating an Install CD for Mobile Clients
 - How to Create an Offline CD for Mobile Clients, 13-1
 - Installing a Mobile Client from the CD, 13-2
 - How to Deploy, 12-1
 - Overview, 12-1
 - Using the Mobile Client for Web-to-Go
 - Client Side Configuration for Secure Socket Layer (SSL), 14-3
 - Installing the Mobile Client for Web-to-Go, 14-1
 - Overview, 14-1
 - Support for Non-SSL Mobile Clients, 14-4

E

- External Authentication
 - External Authentication Code Sample, E-2
 - Using External Authentication, E-1

G

- Getting Started
 - The General Flow, 2-1

I

- Introduction
 - Mobile Server, 1-1

J

- Job Scheduler
 - Administering Scheduled Jobs, 9-5
 - Overview, 9-1
 - Managing the Job Engine, 9-2

K

- keystore
 - creating, 17-1
 - example, 17-2
 - keytool utility, 17-1
 - tester, 17-1

keytool utility, 17-1

M

Managing Devices

- Administering Devices, 10-19
- Architecture, 10-4
- Device Management Client, 10-6
 - Device Management Agent, 10-6
 - Software Update Client, 10-8
 - Software Version Checker, 10-9
- Device Management Server, 10-14
 - Managing Device Commands, 10-16
 - Managing Device Configuration, 10-17
 - Managing Device Platforms, 10-15
 - Managing Network Providers, 10-18
- Mobile Manager, 10-15
 - Overview, 10-15
- Overview and Architecture, 10-1
- Security, 10-5
- Software Management Client, 10-9
 - Installation Configuration File, 10-9
 - Software Installation, 10-9
- Terms and Definitions, 10-2

Managing Synchronization

- Monitoring Synchronization, 8-22
- Overview, 8-1
- The Data Synchronization Manager
 - Introduction, 8-1
- The Message Generator and Processor (MGP), 8-20

Mobile Manager

- Logging On, 3-1

Mobile Manager Overview, 3-2

- Mobile Devices, 3-5
- Mobile Servers, 3-3

Mobile Server

- configuring SSL, 17-2
- Getting Started
 - The General Flow, 2-1
- SSL, 0-xv, 17-1

Mobile Server Configuration Parameters

- CONSOLIDATOR, B-9
- DEBUG, B-5
- FILESYSTEM, B-4
- PUBLIC, B-8
- SERVLET_PARAMETERS, B-8
- WEBTOGO, B-1

Modifying Application Properties, 4-2

Modifying Registry Entries, 4-6

O

Offline Instantiation

- Deploying on Client Machines, 16-6
- Overview, 16-1
- Setting Up the Mobile Client (Distribution) Root Directory, 16-2
- Setting Up the Mobile Client Root Directory, 16-1
- Setting Up the Mobile Server Host and Mobile

Development Kit Host, 16-1

The OLI CD's Directory, 16-2

The OLI Configuration File

Oli.ini, 16-3

Using the OLI Engine, 16-5

Oracle Database Lite 10g

- Concepts and Terminology, 1-2
- Conventions, 1-2
- Detailed Instructions, 1-2
- Implementation Tutorial, 1-2
- What is the Mobile Server, 1-1

P

Provisioning Mobile Applications

- Managing Access Privileges, 6-1
- Managing Snapshots, 6-3

R

Read-Only Snapshots, 6-4

Refreshing a Snapshot, 6-4

Resuming Mobile Applications, 4-4

Running the Mobile Server in Debug Mode, 11-4

S

Scripting Language for the Mobile Server

- Description of the Syntax, D-1
- Examples, D-5
- Running a Script INI File, D-5

Selecting Application Files for Public Use, 4-7

Snapshot Template Variables, 6-4

SSL, 0-xv, 17-1

- configuring Mobile Server, 17-2
- creating certificate, 17-1
- creating keystore, 17-1
- disabling, 17-4
- mixing secure and non-secure sites, 17-4
- No available certificate, 17-4
- troubleshooting, 17-4

Suspending Mobile Applications, 4-4

Synchronization

- Administering the Synchronization Service, 8-9
- Administering the Synchronization Service
 - Managing Configuration Parameters, 8-10
 - Managing Trace Settings and Trace Files, 8-10
- Browsing the Sync Repository, 8-15
- Displaying Operating System (OS) and Java Virtual Machine (JVM) Information, 8-6
- Managing Active Sync Sessions, 8-5
- Managing the Session History List, 8-5
- Managing the Sync Service, 8-2
- Monitoring and Analyzing Sync
 - Performance, 8-7
 - Analyzing Performance of Publications, 8-8
- Scheduling MGP Cycles to Run Inside the Mobile Server, 8-19

System Catalog Views

- Mobile Server Catalog Views, G-1

T

The Consolidator Client API (OCAPI)

 Sync Client Parameters, J-1

Tracing

 Editing the webtogo.ora File, 11-3

 Setting Trace Parameters, 11-2

 Mobile Server Modes, 11-3

Troubleshooting

 Inspecting Files in the Mobile Server

 Repository, A-1

Troubleshooting FAQs

 Accessing Schema Changes to a Published

 Application, A-4

 Accessing the Client Database Offline, A-3

 First Synchronization Causes Browser to Time

 Out, A-3

 Running the Mobile Server in Debug Mode, A-2

U

Updatable Snapshots, 6-4

Using External Authentication, E-1

Using the JDBC Driver

 Upload the Oracle JDBC Driver, H-1

Using the JDBC Thin Driver

 Change the Applet Tag, H-1

W

Mobile Server Configuration Parameters

 , B-1

