Oracle® Database Lite

Administration and Deployment Guide

10*g* (10.2.0)

Part No. B15921-01

June 2005

Oracle Database Lite Administration and Deployment Guide 10*g* (10.2.0)

Part No. B15921-01

Contents

4 Managing Your Mobile Applications

5 Managing Users and Groups

6 Managing Synchronization

7 Job Scheduler

8 Manage Your Devices

9 Offline Instantiation

11 Using the Application Server OID With Mobile Server

12 Manage Security in Oracle Database Lite

13 Performance

14 Configure for National Language Support (NLS)

15 Backup and Recovery

16 Tracing and Logging

17 Reports

18 Adding Popular URLs as Bookmarks to Mobile Server Main Page

A Troubleshooting FAQs

B Configuration Parameters for the WEBTOGO.ORA File

C Data Synchronization Requirements in INIT.ORA

D Scripting Language for the Mobile Server

Glossary

Index

Send Us Your Comments

Oracle Database Lite Administration and Deployment Guide 10*g* (10.2.0)

Part No. B15921-01

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

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

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

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

Preface

This preface introduces you to the *Oracle Database Lite 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. Accessibility 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 more information, visit the Oracle Accessibility Program Web site at

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

Accessibility of Code Examples in Documentation

Screen readers 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, some screen readers 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.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Structure

This guide includes the following topics:

- Chapter 1, "Using the Workspace for Managing the Mobile Server"

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

- Chapter 2, "Installing Mobile Clients"

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

- Chapter 3, "Managing Your Mobile Client"

  This chapter describes how to manage the Mobile clients.

- Chapter 4, "Managing Your Mobile Applications"

  This chapter discusses how to manage Mobile applications.

- Chapter 5, "Managing Users and Groups"

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

- Chapter 6, "Managing Synchronization"

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

- Chapter 7, "Job Scheduler"

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

- Chapter 8, "Manage Your Devices"

  This chapter describes how to manage devices.

- Chapter 9, "Offline Instantiation"

  This chapter discusses the Offline Instantiation feature.

- Chapter 10, "Manage Your Branch Office"

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

- Chapter 11, "Using the Application Server OID With Mobile Server"

  This chapter describes how to use OID, instead of the Mobile Repository.

- Chapter 12, "Manage Security in Oracle Database Lite"

  Oracle Database Lite 10*g* 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.

- Chapter 13, "Performance"

  Consider options for tuning your applications for better performance.

- Chapter 14, "Configure for National Language Support (NLS)"

  This chapter describes NLS configuration for different languages.

- Chapter 15, "Backup and Recovery"

Create a backup strategy and recover your applications and data when necessary.

- Chapter 16, "Tracing and Logging"

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

- Chapter 17, "Reports"

  This chapter enables users to manage the Mobile Server.

- Chapter 18, "Adding Popular URLs as Bookmarks to Mobile Server Main Page"

  Create URLs as bookmarks to any Web address and place on the Mobile Workbench screen.

- Appendix A, "Troubleshooting FAQs"

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

- Appendix B, "Configuration Parameters for the WEBTOGO.ORA File"

  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, "Data Synchronization Requirements in INIT.ORA"

  This appendix describes the Data Synchronization 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, "Bypassing a Proxy Server"

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

- Appendix F, "System Catalog Views"

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

- Appendix G, "Using the JDBC Thin Driver"

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

- Appendix H, "POLITE.INI Parameters"

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

1

Using the Workspace for Managing the Mobile Server

Use the Mobile Workspace GUI tool for managing the Mobile Server; formally known as the Mobile Manager. In addition, Mobile clients on a Windows platform can use the Mobile Workspace GUI for managing Mobile clients with Web-to-Go, BC4J, or Branch Office Mobile clients.

> **Note:** See Chapter 3, "Managing Your Mobile Client" for details on how to manage and synchronize each of the Mobile clients—including Linux, PocketPC, Palm, and Win32—which are not managed using the Mobile Workspace GUI tool.

The Mobile Workspace only displays the relevant functionality depending on who logs in. Use the Mobile Manager to manage Mobile applications; use the Mobile Client Workspace to manage the application on the client-side device. The following sections detail how to use the Mobile Workspace for the Mobile Server. See Chapter 3, "Managing Your Mobile Client" for directions on how to use the Mobile Workspace for your Mobile Client.

- Section 1.1, "Using the Mobile Workspace"
- Section 1.2, "Using Mobile Manager to Manage Your Mobile Server"

1.1 Using the Mobile Workspace

The Mobile Workspace provides you access to your Mobile applications through hyperlinks in a Web browser. The following tabs are available when you use the Mobile Workspace for either managing the Mobile Server or a Web-to-Go client: Applications, Configuration, Help, Sync, and Log Off. See Section 3.1.1, "Instructions for Using the Mobile Workspace" for a description of the functions in each tab—especially the Configuration tab, as it enables you to configure options for your Workspace.

1.2 Using Mobile Manager to Manage Your Mobile Server

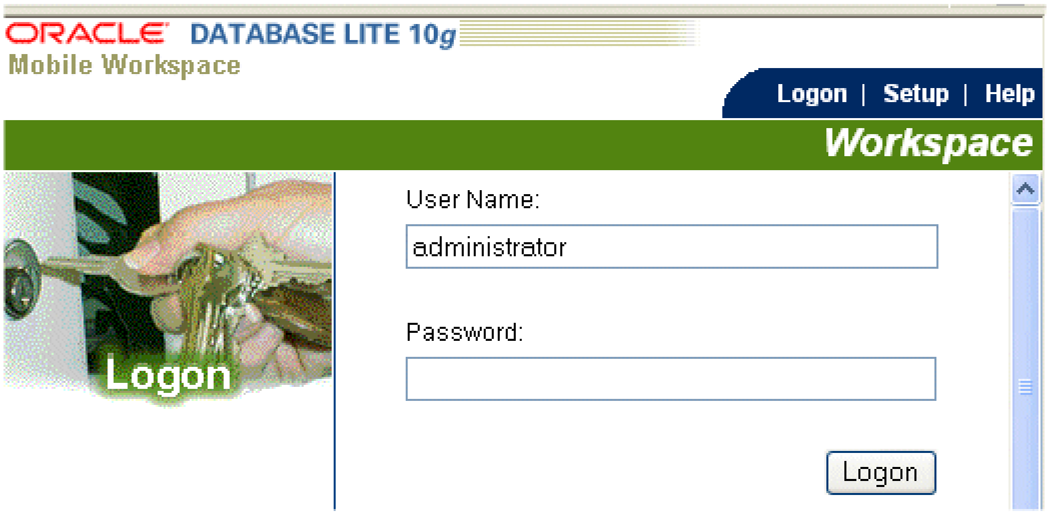
Mobile Manager is used to manage the Mobile Server. An administrator is the only user that is able to log on and use the Mobile Manager. To logon to the Mobile Manager, perform the following steps.

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

   ```
   http://<your_Mobile_Server_host_name>/webtogo
   ```

As Figure 1–1 displays, the Mobile Server Workspace appears with Mobile Server logon page.

Figure 1–1 Logon Page



User Name: administrator

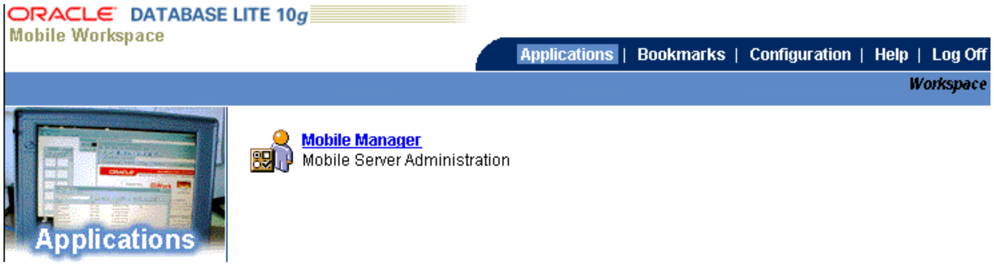
Password:

2. Log on to the Mobile Manager with an administrator username and password. A default administrator is created when you install with username/password of `administrator/admin`. Change the default password or create your own administrator user with appropriate username/password.

   As shown in Figure 1–2, the Mobile Workspace displays the Mobile Manager in the Applications tab, which is the application available to administrators for managing any Mobile Server.

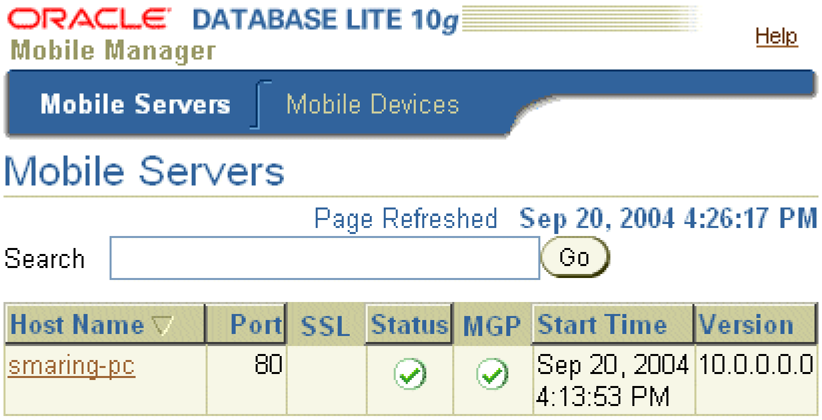
   > **Note:** Section 1.1, "Using the Mobile Workspace" describes the functions of each of the tabs at the top of the Mobile Workspace.

Figure 1–2 Mobile Manager Workspace Page


ORACLE DATABASE LITE 10g
Mobile Workspace
Applications | Bookmarks | Configuration | Help | Log Off
Workspace
Applications
Mobile Manager
Mobile Server Administration

3. Click the **Mobile Manager** icon or link. As Figure 1–3 displays, the Mobile Server farm page appears with a list of installed Mobile Servers that use this repository.

Figure 1–3 Mobile Server Farm Page



Page Refreshed Sep 20, 2004 4:26:17 PM

Host Name ▽	Port	SSL	Status	MGP	Start Time	Version
smaring-pc	80		✓	✓	Sep 20, 2004 4:13:53 PM	10.0.0.0.0

A Mobile Server farm is a group of Mobile Servers configured to run against the same repository. The Mobile Server farm page contains Mobile Server and Mobile Devices tabs. The following sections describe each component of the Mobile Server Farm. Topics include:

- Section 1.2.1, "Viewing Mobile Servers"
- Section 1.2.2, "Viewing Mobile Devices"

1.2.1 Viewing Mobile Servers

The Mobile Servers tab lists all Mobile Servers that are configured to run against the same repository. 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 (see Figure 1–3).

If a Mobile Server instance is not running (status column displays down), the hyperlink for the host name—where the Mobile Server exists—is not enabled. Refresh this page after a Mobile Server instance is started or stopped by clicking on Mobile Server link to see the updated status for the Mobile Server. Using search criteria based on host name, you can filter the Mobile Server display list for only those servers you are interested in.

The following sections describe the Mobile Manager components.

- Section 1.2.1.1, "Mobile Manager Home Page"
- Section 1.2.1.2, "Manage Applications"
- Section 1.2.1.3, "Manage Users"
- Section 1.2.1.4, "Mobile Server Administration"

1.2.1.1 Mobile Manager Home Page

The Mobile Manager home page displays 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. See Chapter 6, "Managing Synchronization" for more information on Data Synchronization and the queues.

- 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.

See Chapter 6, "Managing Synchronization" for more information on Data Synchronization. See Chapter 7, "Job Scheduler" for more information on the Job Scheduler.

1.2.1.2 Manage 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.

See Chapter 4, "Managing Your Mobile Applications" for more information on how to manage your applications.

1.2.1.3 Manage Users

The Users page enables the Mobile Server administrator to manage groups and users and their permissions. See Chapter 5, "Managing Users and Groups" for full details.

1.2.1.4 Mobile Server Administration

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

1. View the sessions that are active. See Section 17.2, "Viewing Active User Sessions" for full details.
2. Edit trace settings. See Chapter 16, "Tracing and Logging" for full details.
3. Edit the configuration file. We prefer that you modify the configuration using the GUI tool; however, if you decide to edit the `webtogo.ora` file directly, you can access it with this link. See Appendix B, "Configuration Parameters for the WEBTOGO.ORA File" for details on the parameters.
4. Add bookmarks. See Chapter 18, "Adding Popular URLs as Bookmarks to Mobile Server Main Page" for full details.
5. View a summary of the database, JRE, and Operating System. See Section 17.1, "Viewing System Status Reports for the Server" for full details.

1.2.2 Viewing 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 briefly describe the functionality available to you in the Mobile Devices tab:

- Section 1.2.2.1, "Installed Mobile Devices"

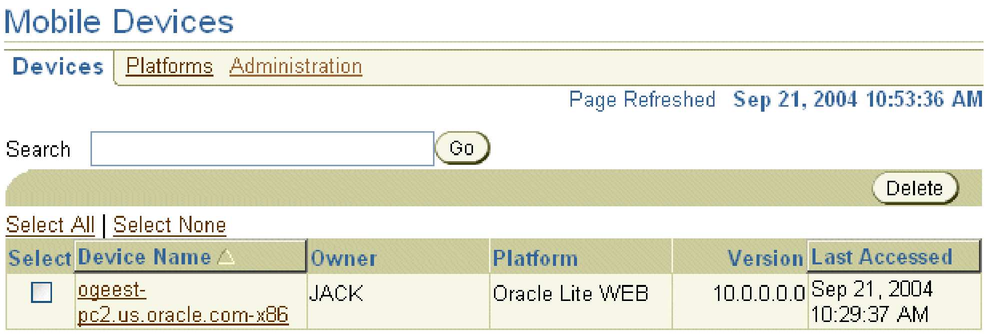
- Section 1.2.2.2, "Mobile Device Platforms"

For full details on managing your Mobile devices, see Chapter 8, "Manage Your Devices".

1.2.2.1 Installed Mobile Devices

View the installed Mobile device information such as device name, owner, platform, version, and date and time on which it was last accessed. Figure 1–4 displays the Devices page.

Figure 1–4 Devices Page



Page Refreshed Sep 21, 2004 10:53:36 AM

Select	Device Name △	Owner	Platform	Version	Last Accessed
□	ogeest-pc2.us.oracle.com-x86	JACK	Oracle Lite WEB	10.0.0.0.0	Sep 21, 2004 10:29:37 AM

To manage the Mobile device, click the Device Name link from the list. For full details on managing your Mobile devices, see Chapter 8, "Manage Your Devices".

1.2.2.2 Mobile Device Platforms

This page lists Mobile device platform information such as platform name, language, enabled, bootstrap, device count, and base platform.

There are hyperlinks that enable you to do the following:

- View device, installed Oracle Database Lite software, back-end database information, what is currently in the command queues, and the logs.
- Extend and manage device platforms—You can extend existing platforms for your own customization—adding other binaries to download or instructions on how to modify the client environment.
- Create commands to be sent to the Mobile device—You can create commands that execute on the device. These commands can start a synchronization, retrieve information, modify the client environment, and many other options.

For full details on managing your Mobile devices, see Chapter 8, "Manage Your Devices".

2 Installing Mobile Clients

This document details how to install Mobile Client software on your client machine, download the application, and synchronize to the Mobile Server Repository to pull down the data. This information is covered in the following sections:

- Section 2.1, "Preparing the Device for a Mobile Application"
- Section 2.2, "Installing the Mobile Client Software"
- Section 2.3, "Configuring the Client for Secure Socket Layer (SSL)"
- Section 2.4, "Using Offline Instantiation to Distribute Multiple Mobile Clients"

See Chapter 3, "Managing Your Mobile Client" on how to perform certain functions from the client. See Chapter 1, "Using the Workspace for Managing the Mobile Server" for information on how to manage functionality from the Mobile Server.

2.1 Preparing the Device for a Mobile Application

In order for a device to execute Mobile applications, you must do the following:

> **Note:** Install the Mobile client for any application after the application is published.

1. Install the Oracle Database Lite Mobile client software that is appropriate to the client platform on your client machine. For example, install the Mobile Client for Web-to-Go or the Mobile Client for Win32 on a Windows 32 client machine.

   See Section 2.2, "Installing the Mobile Client Software" for a full description.

2. Download the user applications and its associated data.

   Synchronize the Mobile client for the first time. When you sign in with the username/password of the user who owns this Mobile client, the Mobile applications to which this user has access are downloaded. In addition, the data for each application is retrieved.

   > **Note:** For more information about synchronization, see Chapter 6, "Managing Synchronization".

3. You can now launch your applications from your client machine or from your Mobile device.

2.2 Installing the Mobile Client Software

Before you install the Mobile Client on your device, make sure that there is 1 MB of space available to download the `setup.exe`. To install the Mobile client software, perform the following tasks.

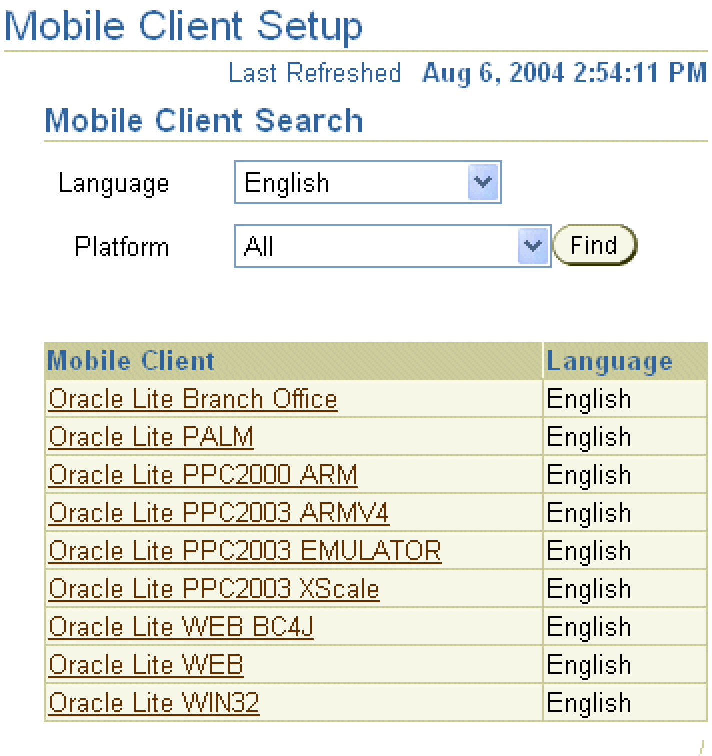
1. On the client, open a browser to point to the Mobile Server using the following URL.

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

   > **Note:** Substitute `https` if using HTTP over SSL.

   Figure 2–1 displays the Mobile Client Setup page, which contains links to install Mobile Client software for multiple platforms and languages. You can select another language than English on the Language pulldown. For viewing platforms, you can choose to see all available platforms for the indicated language, or only those platforms for Windows, Windows CE, or Palm.

Figure 2–1 Mobile Client Setup Page



Mobile Client Setup

Last Refreshed Aug 6, 2004 2:54:11 PM

Mobile Client Search

Language: English

Platform: All

Mobile Client	Language
Oracle Lite Branch Office	English
Oracle Lite PALM	English
Oracle Lite PPC2000 ARM	English
Oracle Lite PPC2003 ARMV4	English
Oracle Lite PPC2003 EMULATOR	English
Oracle Lite PPC2003 XScale	English
Oracle Lite WEB BC4J	English
Oracle Lite WEB	English
Oracle Lite WIN32	English

> **Note:** Available clients may differ. See Section 8.4.4.1, "Enable a Platform for Your Mobile Client" for more information.

2. Click the Mobile client for your language and client platform.
3. The Save As dialog box appears. The file name field displays the setup executable file for the selected platform as an `.exe` file type. Save the executable file to a directory on the client machine.

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 perform a `hotsync` to install the Mobile Client for Palm.

For WinCE, you install any of the Oracle Lite Pocket PC platforms to ActiveSync. Then, when the device is put into the cradle, ActiveSync installs the Oracle Database Lite onto the device when it synchronizes.

4. Install the Mobile client. For all platforms, except Palm and installing WinCE on ActiveSync, go to the directory where you saved the setup executable file. Double-click the file to execute it.
5. Enter the username and password for the Mobile client user.
6. Provide the directory name where to install the Mobile client.
7. Once installed, synchronize the Mobile client for the first time. During the first synchronization, all applications and data for this user is brought down and installed on your Mobile client. See Table 2–1 for a description of synchronization for each platform.

Note: See Section 2.2.1, "Configuring for Default Sync When Installing the Client" for directions on how to enable a default synchronization after any client installation on your device.

Table 2–1 Initializing the First Synchronization for Each Mobile Client Platform

Oracle Mobile Client	Initial Synchronization Details
Oracle Lite Web (Web-to-Go)	The synchronization step takes place when you click **Next**, after executing the `setup.exe`. This prompts you to login to the Mobile Client for Web-to-Go. If you want to synchronize at another time, do the following: 1. Open a browser to the Mobile client. For example, if you install a Web client with port 8080, point the browser to `http://localhost:8080/webtogo`. 2. Log in with your username/password. 3. Click **Sync** on the tabs in the upper right corner.
Oracle Lite Palm	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 perform a `hotsync` to install the Mobile Client for Palm.
Oracle Lite PocketPC for WinCE devices	If you install the PocketPC platform to ActiveSync, insert the WinCE device in the cradle. ActiveSync performs a synchronization to install Oracle Database Lite on the device. After Oracle Database Lite is installed on the device, then start the Device Manager Agent on the device by executing `dmagent.exe`, which is in the `orace` directory.

Table 2–1 (Cont.) Initializing the First Synchronization for Each Mobile Client Platform

Oracle Mobile Client	Initial Synchronization Details
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**.

2.2.1 Configuring for Default Sync When Installing the Client

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

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

See Section 8.6, "Sending Commands to Your Mobile Devices" for more details on sending commands to your Mobile device.

2.3 Configuring the Client 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. A complete description of how to configure your Mobile client to use SSL is described in Section 12.1.4, "Client-Side Configuration for Secure Socket Layer (SSL)".

2.4 Using Offline Instantiation to Distribute Multiple Mobile Clients

You can enable your users to install their client using a distribution method, such as a CD, through the network, or email. To install the Mobile client and perform the first synchronization to retrieve the applications (with the initial data) can be a performance issue. In this case, the administrator pre-creates the Mobile binaries with the user ODB files (includes the applications and data for the user) to the client. The download of this package is faster than having each user perform the first synchronization on their device. Thus, this procedure helps users avoid an expensive performance hit when creating and synchronizing the Mobile client for the first time.

Offline instantiation is a tool that enables an administrator to gather and package the Mobile client binaries and the user applications and data into a single directory. Offline instantiation is part of the Mobile Development Kit, which can be installed only on a Windows platform. Thus, you create all of your user distribution files on a Windows machine and you can only create multiple user distribution files for Web-to-Go, Branch Office, Win32, and WinCE Mobile clients. We recommend that you use the same Windows environment where a Mobile server exists to create your distribution files.

See Chapter 9, "Offline Instantiation" for full instructions on how to use the Offline Instantiation engine to create and deploy multiple clients.

3 Managing Your Mobile Client

The types of Mobile clients are as follows:

- Web clients (such as Web-to-Go, Branch Office, and BC4J): The application built for these clients uses a Java browser. You can initiate the application or synchronization using the Mobile Workspace GUI. See Section 3.1, "Using the Mobile Client Workspace to Manage Your Web Clients" for more information.
- Linux, Win32, and PocketPC clients: These applications are client/server applications. Thus, start the application as you would start any application on these platforms. You can initiate synchronization either by implementing it within your application using the synchronization APIs (see the *Oracle Database Lite Developer's Guide* for more information) or by executing the `msync` executable, described in Section 3.2, "Using the msync GUI to Initiate Synchronization of Your Linux, PocketPC, and Win32 Client/Server Application Clients".
- Palm clients: Initiate synchronization with the synchronization button on the Palm pilot.

There are several tools that you can use on the client to manage functionality. The following sections describe other tools that you can use in each platform.

- Section 3.3, "Using the Device Management Client GUI to Manage the Client-Side Device"
- Section 3.4, "Initiating Updates of Oracle Database Lite Software from the Client"

3.1 Using the Mobile Client Workspace to Manage Your Web Clients

The Mobile Workspace GUI tool is used for both the Mobile Manager (of the Mobile Server) and for managing Web-to-Go, BC4J, or Branch Office Mobile clients. The Mobile Workspace only displays the relevant functionality for the user that logs in. Use the Mobile Manager to manage Mobile applications; use the Mobile Client Workspace to manage the application on the client-side device. The following sections detail how to use the Mobile Workspace for your Web clients. See Chapter 1, "Using the Workspace for Managing the Mobile Server" for information on how to use the Mobile Workspace for managing the Mobile Server.

- Section 3.1.1, "Instructions for Using the Mobile Workspace"
- Section 3.1.2, "Log on to Mobile Client Workspace"
- Section 3.1.4, "Executing Mobile Applications Installed on Your Mobile Client"
- Section 3.1.5, "Scheduling Data Synchronization Jobs"

3.1.1 Instructions for Using the Mobile Workspace

The Mobile Workspace provides you access to your Mobile applications through hyperlinks in a Web browser. The following tabs are available when you use the Mobile Workspace for a Web client:

- Section 3.1.1.1, "Applications Tab"
- Section 3.1.1.2, "Configuration Tab"
- Section 3.1.1.3, "Help Tab"
- Section 3.1.1.4, "Sync Tab"
- Section 3.1.1.5, "Log Off Tab"

3.1.1.1 Applications Tab

The Applications tab displays the list of applications that have been installed on the client, which you can execute. To access a Mobile application, click the icon or application name. The Mobile Workspace allows you to execute multiple applications concurrently in separate browsers.

> **Note:** When you log on as an Administrator, the Mobile Manager is listed as the available application.

3.1.1.2 Configuration Tab

To modify your Workspace configuration, select the Configuration tab.

> **Note:** When you select the Configuration tab, maximize the window to your display. If you do not, you might not be able to see all of the configuration options on the left.

The following options appear on the left:

- Workspace Settings: Configure your Mobile Workspace settings, such as display options and Web-to-Go client options. The following discuss the Web-to-Go settings:
  - Setting default settings for synchronization applies only to the offline mode. If you select this option, then all applications are synchronized at once. If you do not select this option, then when you synchronize, Web-to-Go presents you with a list of applications for you to select which ones to synchronize. Thus, you can reduce sychronization time by selecting which applications you want to synchronize. Web-to-Go notifies you when new applications become available. Mobile Client for Web-to-Go Mode. This label applies only to the offline mode. Selecting **Always Offline** enables you to work continuously in offline mode. Selecting the **On/Offline** option allows you to switch between the offline and online modes. See Section 3.1.1.2.1, "Offline Mode Options" for full details on offline and online modes.
  - Ask before ugrading the Mobile Client for Web-to-Go? This label applies only to the offline mode. Selecting this option instructs the Mobile Client for Web-to-Go to ask you before downloading the newer versions of the Mobile Client for Web-to-Go software that the Mobile Server detects in the Mobile Server Repository. If you do not select this option, your Mobile Client for Web-to-Go will be upgraded with the new version of the client the next time

you sync if a new version of the Mobile Client for Web-to-Go is available—if configured to do so within the Mobile Manager.

- Application Settings: Select the applications that you want to synchronize by default. In the default synchronization mode, the Mobile Server synchronizes your client applications automatically with the Mobile Server when you go online. See Section 3.1.1.2.2, "Configuring Application Synchronization Default" for full details.
- Change Password
- Create a Data Synchronization Job: You can schedule a time when synchronization is to be initiated on the client. This is configured as a job initiated by the client. There is a small job engine on the client, so when you set up a job to execute at a certain time and interval, it initiates the application at the specified time. The only job that you can schedule on the client is for synchronization. For more information on scheduling jobs on the server-side, see Chapter 7, "Job Scheduler".

  Scheduled replication jobs will not run on client if there are any pending transactions that have not been committed on that client.

  > **Note:** You can only schedule jobs for your own sites when the Mobile Client for Web-to-Go is in offline mode. For more information, see Section 3.1.1.2.1, "Offline Mode Options".

- In addition, you can disable or enable remote access for the Mobile client. See Section 3.1.1.2.3, "Enable or Disable Remote Access for Mobile Client" for more information.

3.1.1.2.1 Offline Mode Options The Mobile Client for Web-to-Go mode options enable you to switch from **Always Offline** (the default value) to the **On/Offline** setting, where you can switch between the online and offline modes.

In the online mode, your client is connected to the Mobile Server. A synchronization starts the process where your data changes are merged into the application tables in the Oracle database. Switching to the offline mode disconnects you from the Mobile Server; thus, a synchronization cannot occur where updates are provided by the client. Updates can only be provided when the client is connected. When you go offline, Web-to-Go can download the latest snapshots of your data and applications, enabling you to work without a connection to the Mobile Server. These updates are finally stored on the device when the device reconnects.

When you select the **On/Offline** option, the Go Online tab replaces the Sync tab. The Go Online tab indicates that you are in offline mode. Clicking the Go Online tab connects you to the Mobile Server. Once the client is connected to the Mobile Server, the Go Offline tab replaces the Go Online tab. Clicking the Go Offline tab disconnects you from the Mobile Server and synchronizes selected applications and data.

> **Note:** Synchronization between the Mobile Client for Web-to-Go and the Mobile Server is asynchronous; if you use the **On/Offline** mode, changes you make just prior to switching between the offline and online modes may not be immediately reflected in the refreshed data set.

3.1.1.2.2 Configuring Application Synchronization Default The Application Settings option allows you to select the applications that you want to assign to the default synchronization setting. In the default mode, Web-to-Go synchronizes your client application automatically with the Mobile Server when you synchronize. Selecting fewer applications decreases the amount of data to download and speeds up the synchronization process.

To select the applications for synchronization, click Application Settings. The application settings page appears in the right frame of the Mobile Workspace and displays the following information:

Table 3–1 describes the Application Settings page.

Table 3–1 Application Settings Page Description

Label	Description
Synchronize	If selected, Web-to-Go automatically synchronizes your client application with the Mobile Server when you synchronize.

Select the Synchronize check box next to the application and click **Save**. If you clear the check box, you receive a notification asking if you wish to synchronize this application before Web-to-Go synchronizes. Clicking **Save** commits the selected applications for automatic synchronization. If you make an error, click **Reset** to return to the previous settings.

3.1.1.2.3 Enable or Disable Remote Access for Mobile Client To block a remote machine from getting access to the Mobile Client for Web or Mobile Client for BC4J, set the `DISABLE_REMOTE_ACCESS` parameter in the client-side `webtogo.ora` file to `YES`. Once this parameter is set to `YES` and Mobile Client is restarted, only the request coming from the local machine is served by the Mobile Client listener. Any other request is blocked and not served.

For Branch Office clients, this parameter must be set to `NO`, as all clients must have remote access. For more information on the `DISABLE_REMOTE_ACCESS` parameter, see Section B.1, "[WEBTOGO]".

3.1.1.3 Help Tab

The Help tab launches the online help system.

3.1.1.4 Sync Tab

If your device is not currently connected, then you can run Mobile applications exclusively in offline mode and synchronize data and applications at any time through the Sync tab. When you select the Sync tab, a synchronization of your data is initiated with the Oracle database. In addition, application updates and new applications accessible to the user are downloaded from the Mobile Server. After synchronization, you can work continuously in offline mode, disconnected from the Mobile Server. You only need a connection to the Mobile Server when you synchronize your data and applications.

For information on setting synchronization options, see Section 3.1.1.2, "Configuration Tab".

3.1.1.5 Log Off Tab

The Log Off tab automatically closes all running applications and returns you to the logon page.

3.1.2 Log on to Mobile Client Workspace

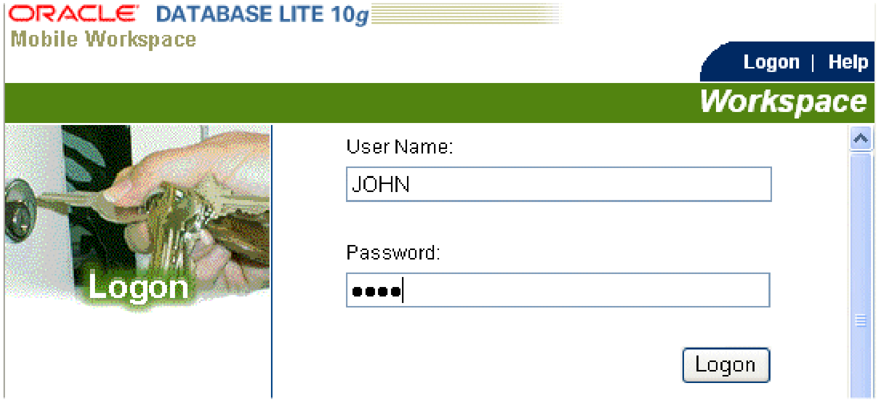
You must have installed the Mobile Client for Windows on your client machine. Then, for any Windows-based clients, you can connect to the Mobile Client Workspace with a browser by connecting to one of the following:

- `http://<client_machine>/webtogo`
- `http://localhost/webtogo`.

Alternatively, you can start the client by double-clicking on the following icon in the corner of your Windows desktop:

Or you can launch it by accessing your Oracle Database Lite program group and choosing Web-to-Go. Figure 3–1 displays the Web-to-Go Logon page.

Figure 3–1 The Mobile Workspace Logon Page



User Name: JOHN

Password: ••••

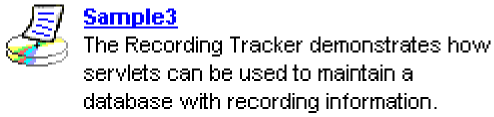
Enter your user name and password and click Logon.

3.1.3 Customizing the Mobile Client Workspace

You can customize the Mobile Client Workspace. See the" Customizing the Workspace Application" section in the *Oracle Database Lite Developer's Guide*.

3.1.4 Executing Mobile Applications Installed on Your Mobile Client

Web-to-Go applications appear in the Mobile Workspace with an icon, name, and description.



Sample3

The Recording Tracker demonstrates how servlets can be used to maintain a database with recording information.

The icon and application name are both hyperlinks. To execute a Web-to-Go application, click either the icon or application name.

Web-to-Go enables you to work in offline mode—that is, disconnected from the Mobile Server. You need a connection to the Mobile Server only when you choose to synchronize any data changes from the client with the Oracle database.

Web-to-Go propagates the tables that your applications use on the Mobile Server to the Mobile Client for Web-to-Go as database snapshots for use in offline mode. You can work continuously in offline mode, with Web-to-Go storing your data changes in the Oracle Lite database. When you click the Sync tab, Web-to-Go updates the Oracle database with any data changes you made on your client in offline mode. The Mobile Server downloads any new applications, application changes, or data changes to your client.

Although you can work continuously in offline mode, you can change from the default offline mode to a setting that enables you to switch between the online and offline modes. For more information, see "Offline Mode Options".

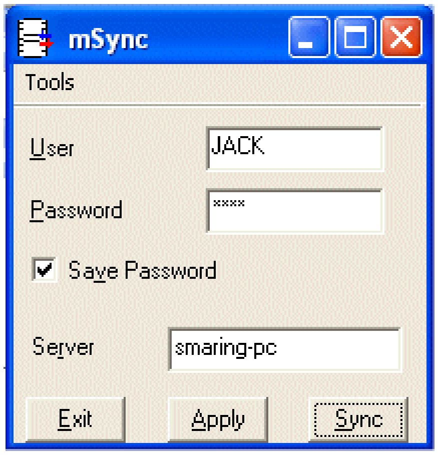
3.1.5 Scheduling Data Synchronization Jobs

In offline mode, the Mobile Workspace enables you to create data synchronization jobs for your site from the Mobile Client for Web-to-Go. This synchronization job automatically triggers synchronization with the Mobile Server at the start date and at the specified time for this job that you set using the Mobile Workspace. See Section 3.1.1.2, "Configuration Tab" for more information.

3.2 Using the msync GUI to Initiate Synchronization of Your Linux, PocketPC, and Win32 Client/Server Application Clients

You can initiate synchronization of the client using the msync GUI, as shown in Figure 3–2.

Figure 3–2 Using the msync GUI to Initiate Synchronization



User JACK

Password ****

☑ Save Password

Server smaring-pc

To bring up the msync GUI, execute `msync.exe` on PocketPC and Win32 or `msync` on Linux, which is located in the `/bin` subdirectory under the directory where you installed the Mobile client. Modify the following supplied values, if incorrect:

- Username and password for the user that is starting the synchronization.
- Check if you want the password saved for future requests.
- Host name where the Mobile Server is installed.

Click **Sync** to start the Synchronization. Click **Apply** to save any modifications you made to the entries. Click **Exit** to leave the tool.

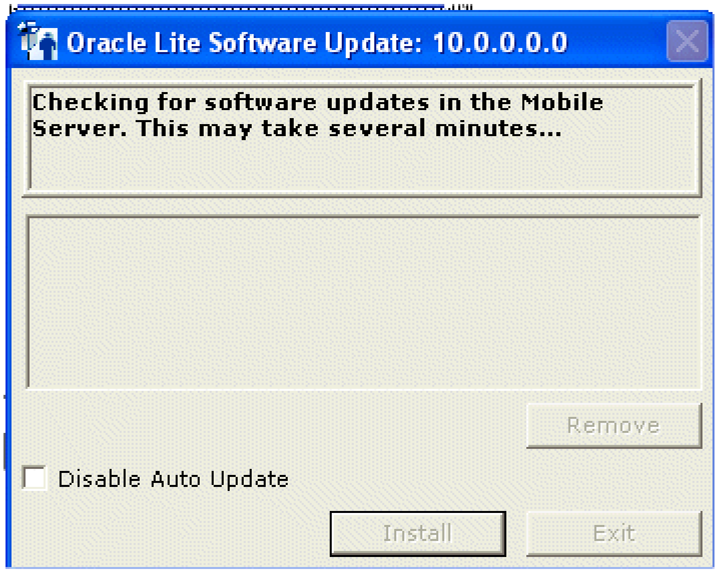
3.3 Using the Device Management Client GUI to Manage the Client-Side Device

On any client—except for the Palm clients—you can manage the Mobile device client software using the Oracle Lite Device Manager. See Section 8.8, "Using the Device Management Client GUI to Manage Device on Client-side" for a full description.

3.4 Initiating Updates of Oracle Database Lite Software from the Client

On Windows clients, you can initiate a request for software updates from the Mobile Server by executing the Oracle Lite Update tool, as shown in Figure 3–3. To execute, choose **Oracle Lite Update** from the Oracle Database Lite Programs list.

Figure 3–3 Updating Oracle Database Lite Software



Checking for software updates in the Mobile Server. This may take several minutes...

□ Disable Auto Update

When updates are located, you can select items that you do not want to update and click **Remove**. When all updates are satisfactory, click **Install**. When you are finished, click **Exit**.

If you check the Disable Auto Update checkbox, then you must execute this tool to receive software updates in the future. You can also disable automatic updates from the Mobile Manager. See Section 8.4.3, "Allowing Software Upgrades to All Mobile Devices in a Platform" for more information.

4

Managing Your Mobile Applications

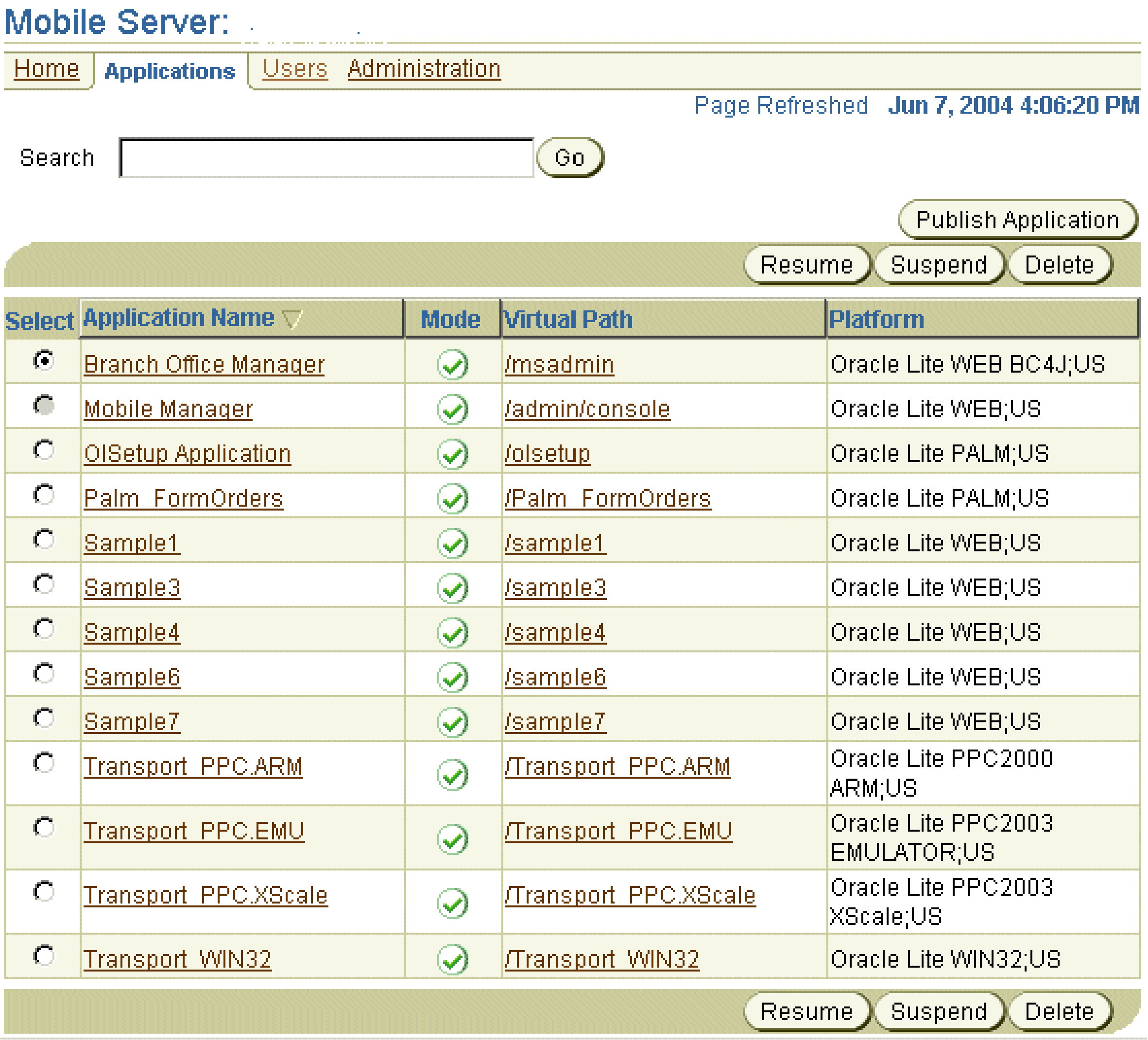
The administrator manages applications through the following tasks:

- Section 4.1, "Listing Applications"
- Section 4.2, "Publishing Applications to the Mobile Server Repository"
- Section 4.3, "Manage Application Properties or Users"
- Section 4.4, "Managing User-Specific Application Parameters (Data Subsetting)"
- Section 4.5, "Managing Access Privileges for Users and Groups"
- Section 4.6, "Selecting Application Files for Public Use"
- Section 4.7, "Adding Web Application Archive (WAR) Files"
- Section 4.8, "Modifying Registry Entries"

4.1 Listing Applications

You can view all applications that are currently published on this Mobile Server from the Mobile Server home page. Click **Applications**. Figure 4–1 displays the Applications page, which lists existing applications and corresponding virtual paths.

Figure 4–1 Applications Page



Page Refreshed Jun 7, 2004 4:06:20 PM

Select	Application Name ▽	Mode	Virtual Path	Platform
●	Branch Office Manager	✓	/msadmin	Oracle Lite WEB BC4J;US
○	Mobile Manager	✓	/admin/console	Oracle Lite WEB;US
○	OlSetup Application	✓	/olsetup	Oracle Lite PALM;US
○	Palm_FormOrders	✓	/Palm_FormOrders	Oracle Lite PALM;US
○	Sample1	✓	/sample1	Oracle Lite WEB;US
○	Sample3	✓	/sample3	Oracle Lite WEB;US
○	Sample4	✓	/sample4	Oracle Lite WEB;US
○	Sample6	✓	/sample6	Oracle Lite WEB;US
○	Sample7	✓	/sample7	Oracle Lite WEB;US
○	Transport_PPC.ARM	✓	/Transport_PPC.ARM	Oracle Lite PPC2000 ARM;US
○	Transport_PPC.EMU	✓	/Transport_PPC.EMU	Oracle Lite PPC2003 EMULATOR;US
○	Transport_PPC.XScale	✓	/Transport_PPC.XScale	Oracle Lite PPC2003 XScale;US
○	Transport_WIN32	✓	/Transport_WIN32	Oracle Lite WIN32;US

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.

4.2 Publishing Applications to the Mobile Server Repository

A developer builds the Mobile application and packages it together with a publication. At this point, the application is ready to be published to the Mobile Server through one of the following options:

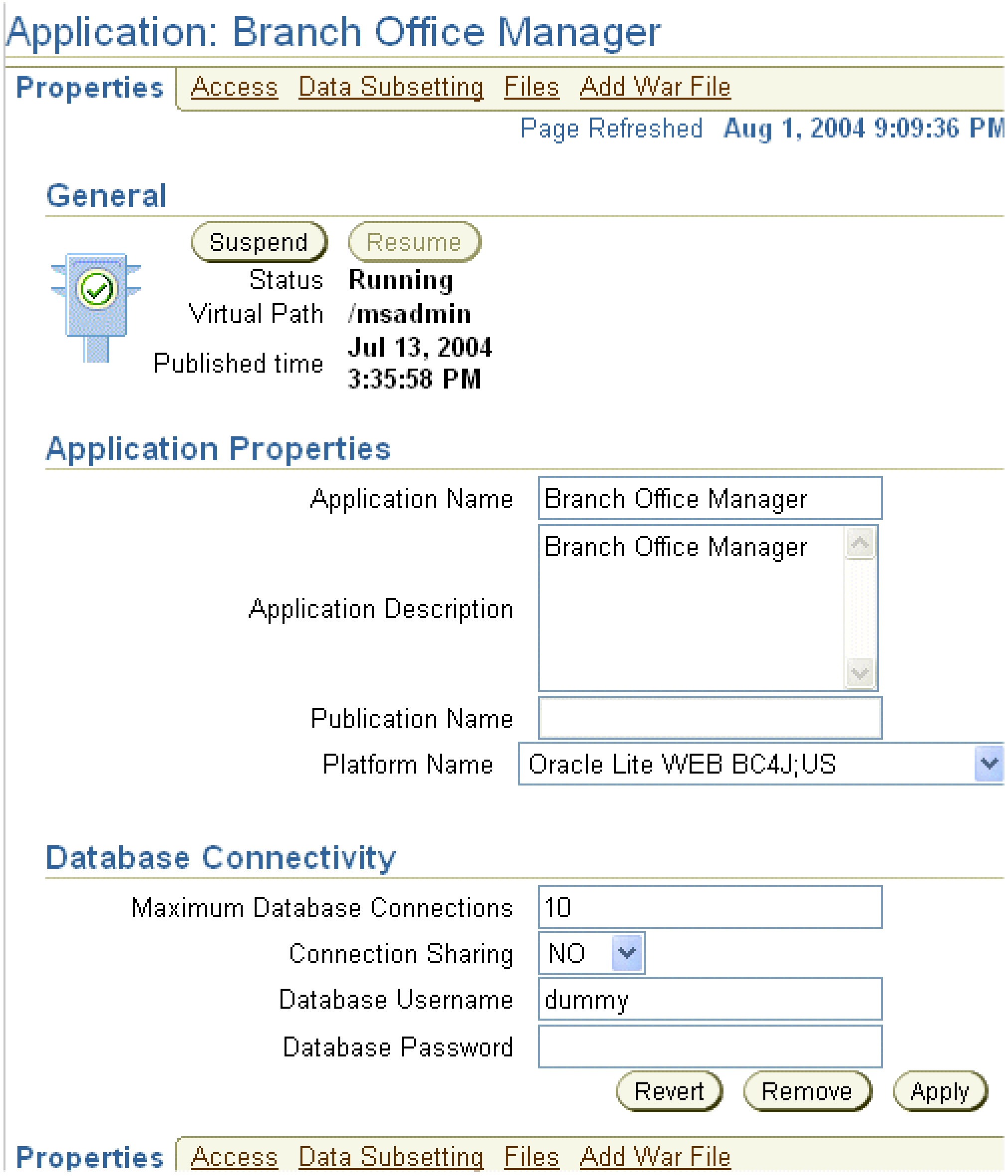
- An organizer or administrator can publish the application using the packaging wizard.
- An administrator can publish the application using the Mobile Manager from the Applications page with the Publish Application button, as shown in Figure 4–1.

To delete any application, select the application and click **Delete**.

4.3 Manage Application Properties or Users

From the Applications page, you can modify application properties for each application. Click on the application name to bring up its Properties page, shown in Figure 4–2.

Figure 4–2 Application Properties Page



Application: Branch Office Manager

Page Refreshed Aug 1, 2004 9:09:36 PM

General

Status: Running
Virtual Path: /msadmin
Published time: Jul 13, 2004 3:35:58 PM

Application Properties

Application Name: Branch Office Manager
Application Description: Branch Office Manager
Publication Name:
Platform Name: Oracle Lite WEB BC4J;US

Database Connectivity

Maximum Database Connections: 10
Connection Sharing: NO
Database Username: dummy
Database Password:

Table 4–1 describes the application properties that you can modify in this screen.

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.

Table 4–1 (Cont.) Application Properties Page Description

Field	Description
Publication Name	Your application is published with a publication that contains the definition of the snapshot data for the clients. This field displays the publication name of the Mobile application. You cannot modify this field.
Platform Name	The platform name consists of the platform type and the language of the application. You can modify this platform to another type as displayed in the pull-down list.

Table 4–2 describes the following data connectivity properties that are available for Web-to-Go applications.

- You can limit the number of connections are allowed to the database. To manage the performance and available resources of the database, you may want to set a limit of how many connections each application can have open at any given time.
- You can enable a connection pool for your Web-to-Go application to use. Connection pools are set up for performance reasons. As each connection request comes in, a connection from the pool is used for the incoming request. When the request ends, the connection is returned to the pool. This eliminates the time necessary for creating and destroying the connections each time a new request comes in.

Table 4–2 Data Connectivity Properties

Property	Descripton
Maximum Database Connections	Number of maximum database connections used by your Mobile application.
Connection Sharing	Select **Yes** if you want to use connection pooling.
Database User Name	Username for the schema used by the application in the database.
Database Password	Password of the schema user.

To retain the modified application properties, click **Apply**. To remove the application, click **Remove**. To reset the Application Properties page, click **Reset**.

4.4 Managing User-Specific Application Parameters (Data Subsetting)

In retrieving data for each user, the application often requires that a parameter is set defining the type of data to retrieve. Set this parameter, also known as data subsetting, in one of two places: on the Data Subsetting page off the Applications page or on the Data Subsetting page off the Users page. See Section 5.3, "Managing Application Parameter Input (Data Subsetting)" for directions on how to manage the input parameter values for the application from the User page.

What is Data Subsetting? When you set up your publication item, you may have set up an input parameter that defines what snapshot of data is to be retrieved for this user. For example, if you have an application that retrieves the customer base for each sales manager, the application needs to know the sales manager’s identification number to retrieve the data specific to each manager. Thus, if you set up each sales manager as a unique user and set their identification number in the data subsetting screen, then the application is provided that unique information for retrieving data.

1. Navigate to the Applications page. Click the specific application.

2. Click **Data Subsetting**. The Data Subsetting page enables the administrator to add parameter input for each user of this application. This displays all of the users that the application is associated with.
3. Select the user for which you want to add the parameter value.
4. Enter the parameter values for the application.
5. Click **Save**.

4.5 Managing Access Privileges for Users and Groups

Similar to Data Subsetting, you can set the access privileges for the application either from the Users page or from the Applications page—except for groups. Groups can only be given access to applications from the Applications page. See Section 5.2.1, "Grant or Revoke Application Access to Users" for directions on how to manage the access privileges from the user page.

The Mobile Server Administrator grants access privileges to Mobile applications by designating the users that can access these applications. This section enables an administrator to grant or revoke application access to users and groups. Topics include:

- Grant Application Access to Users and Groups
- Revoke Application Access to Users and Groups

4.5.1 Grant Application Access to Users and Groups

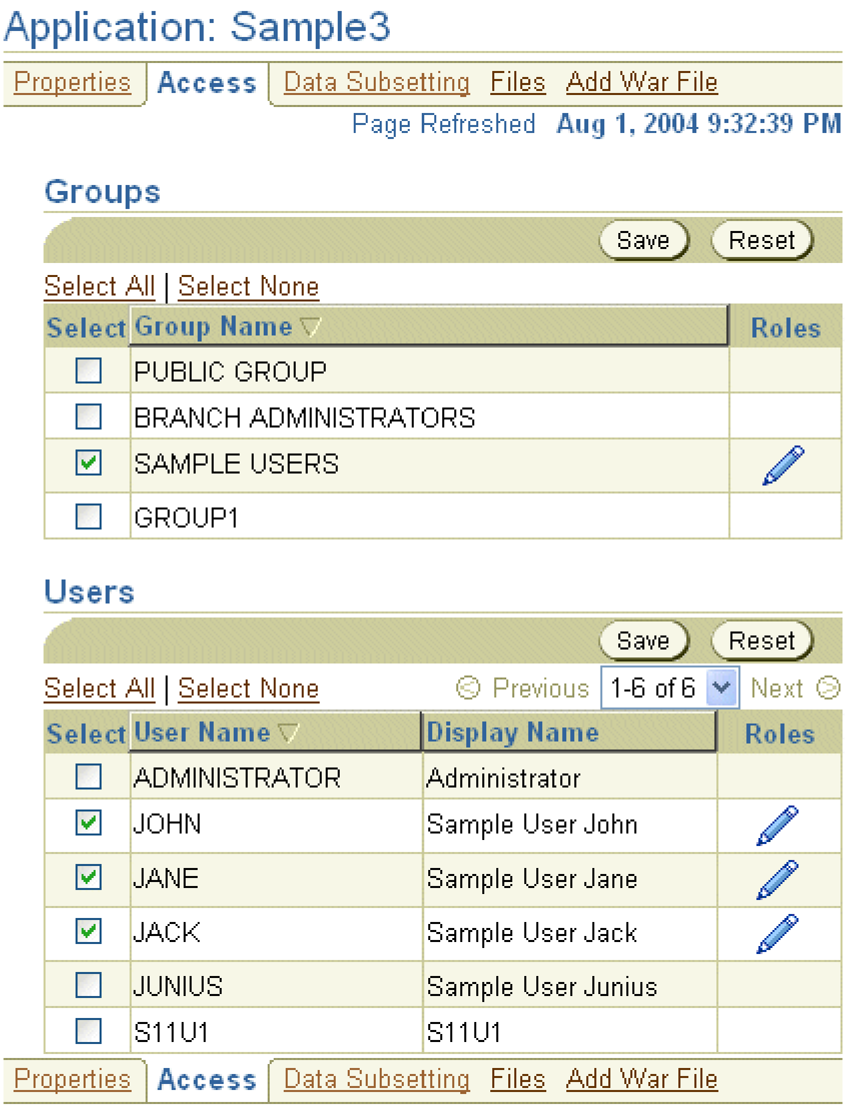
The administrator can grant access to applications for specific users within the Mobile Manager, as follows:

1. Navigate to the Applications page. Click the specific application that you wish to modify. The Properties page appears.
2. Click **Access**. The Access page displays a list of users and groups for this application.
3. Select the checkbox next to each user or group that you wish to give access to for this particular application.
4. Click **Save**.

As Figure 4–3 displays, the Access page displays a list of available users and groups for the Sample3 application. Select the users or groups that you want Sample3 to have access to and click **Save**. In this example, the administrator granted access for the Sample3 application to the SampleUsers group and to the users: John, Jane, and Jack.

Note: Once you provide access to a group, all users in that group have access to this application.

Figure 4–3 Granting Application Access



Application: Sample3

Page Refreshed **Aug 1, 2004 9:32:39 PM**

Groups

Select	Group Name	Roles
□	PUBLIC GROUP	
□	BRANCH ADMINISTRATORS	
☑	SAMPLE USERS	✎
□	GROUP1	

Users

Select	User Name	Display Name	Roles
□	ADMINISTRATOR	Administrator	
☑	JOHN	Sample User John	✎
☑	JANE	Sample User Jane	✎
☑	JACK	Sample User Jack	✎
□	JUNIUS	Sample User Junius	
□	S11U1	S11U1	

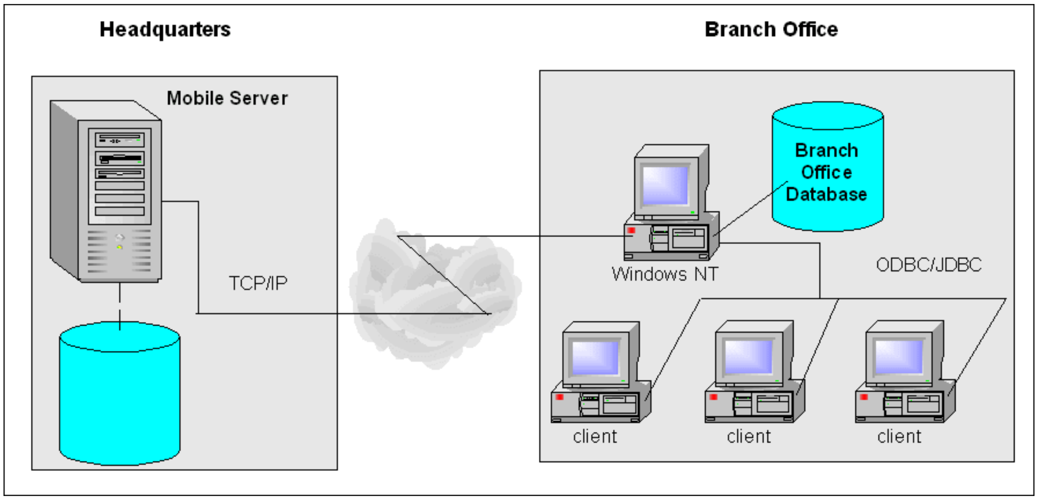
4.5.2 Revoke Application Access to Users and Groups

To revoke application access to any user or group, clear the check box displayed against a group or user name and click **Save**.

4.6 Selecting Application Files for Public Use

To use the Branch Office, you install the Mobile client software on the Branch Office itself. Then, as shown in Figure 4–4, Branch Office maintains its own clients—which are not Mobile clients— by downloading the application onto its clients, which, in turn, communicate directly with the Branch Office. For an overview on what Branch Office is and how to use it, see Chapter 10, "Manage Your Branch Office". For details on setting up a Branch Office and its clients (requiring you to make certain application files public), see Section 10.2, "Branch Office Installation and Configuration".

Figure 4–4 A Branch Office Environment



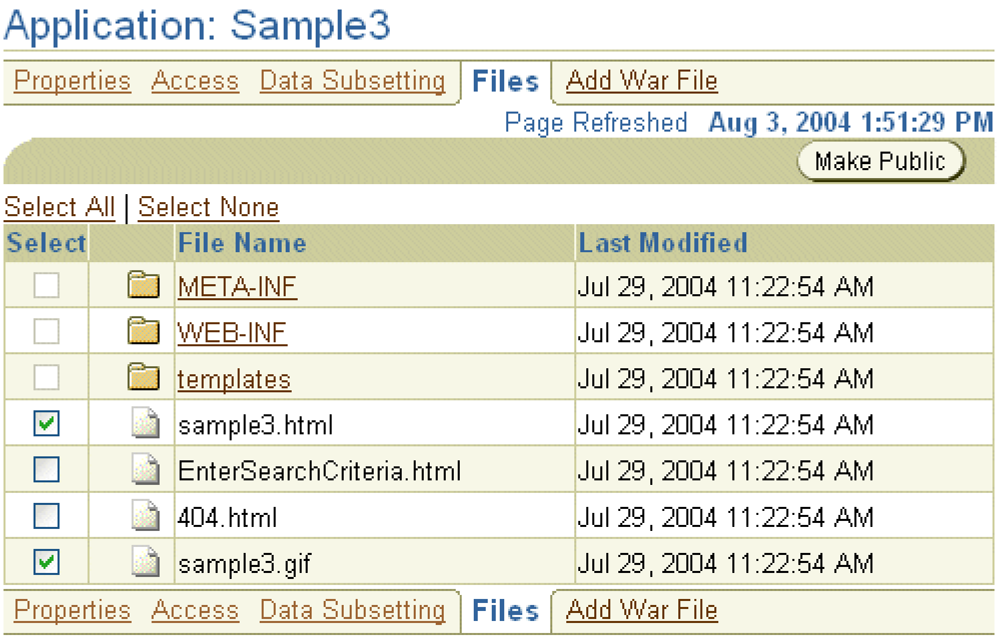
How do you enable the Branch Office to create its clients? The application that is to be executed on the Branch Office clients is published to the Mobile Server. The application, often an executable, that is to be installed on the client is exposed as a public file. The Branch Office downloads the public application executable and is able to install this application on its clients.

Do the following to make the application installation file public.

1. Navigate to the Applications page and click the application link. The Applications home page appears.

   Click **Files**. As Figure 4–5 displays, the Files page lists application files that are assigned for public use.

Figure 4–5 Files Page



Application: Sample3

Properties Access Data Subsetting **Files** Add War File

Page Refreshed **Aug 3, 2004 1:51:29 PM**

Make Public

Select All | Select None

Select		File Name	Last Modified
☐		META-INF	Jul 29, 2004 11:22:54 AM
☐		WEB-INF	Jul 29, 2004 11:22:54 AM
☐		templates	Jul 29, 2004 11:22:54 AM
☑		sample3.html	Jul 29, 2004 11:22:54 AM
☐		EnterSearchCriteria.html	Jul 29, 2004 11:22:54 AM
☐		404.html	Jul 29, 2004 11:22:54 AM
☑		sample3.gif	Jul 29, 2004 11:22:54 AM

Properties Access Data Subsetting **Files** Add War File

2. Select the check box against the application file that you want made public and click **Make Public**

Synchronize the Branch Office, which was previously set up with the appropriate Mobile client software. This brings down the application, the data for this application, and the public application installation file.

On the Branch Office, copy and execute the application public installation file on each Branch Office client. This installs the application on each Branch Office client.

Users can download public files from the Branch Office through the following URL.

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

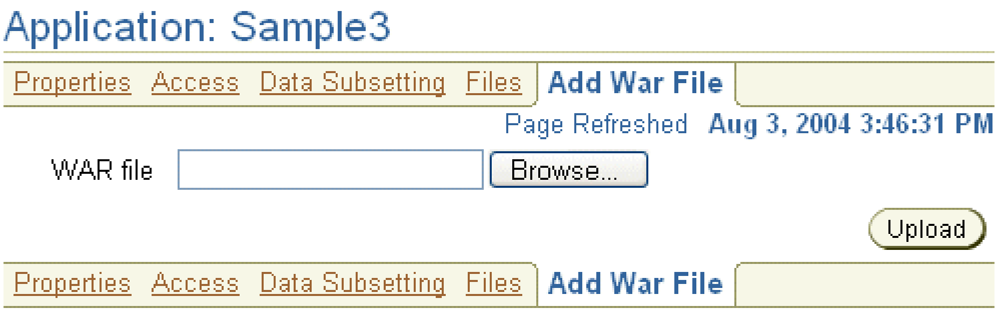
For full instructions and details, see Section 10.2, "Branch Office Installation and Configuration".

4.7 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). 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 4–6 displays, the Add WAR File page appears.

Figure 4–6 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.

4.8 Modifying Registry Entries

If you have used registry entries in the past, you can enable them in the Mobile Manager by adding the `REGISTRY_TAB` parameter to the `webtogo.ora` file. Once you have located the `webtogo.ora` file, enter your registry value in the `REGISTRY_TAB` parameter.

5 Managing Users and Groups

This chapter describes how to manage users and groups using the Mobile Manager. The following topics are covered in this chapter:

- Section 5.1, "Managing Users and Groups"
- Section 5.2, "Managing Access Privileges for Users and Groups"
- Section 5.3, "Managing Application Parameter Input (Data Subsetting)"
- Section 5.4, "Assigning Application Roles to Users"
- Section 5.5, "Creating an Administrator"
- Section 5.6, "Manually Adding Devices for a User"
- Section 5.7, "Set Update Policy for Software Updates for the User"

5.1 Managing Users and Groups

The following sections discuss how to manage users. Topics include:

- Section 5.1.1, "What Are Mobile Server Users?"
- Section 5.1.2, "Displaying Users"
- Section 5.1.3, "Adding New Users"
- Section 5.1.4, "Adding New Groups"
- Section 5.1.5, "Managing OID Users in the Mobile Server"

5.1.1 What Are Mobile Server Users?

The Administrator or User of Mobile Server user types are described in the following sections:

> **Note:** Do not confuse Mobile Server users with database users. Each Mobile Server user is authenticated by the Mobile Server for access to applications and appropriate publications. The Mobile Server users are not used to access data on the database.

- Section 5.1.1.1, "Mobile Server User Privilege: Administrator"
- Section 5.1.1.2, "Mobile Server User Privilege: User"

5.1.1.1 Mobile Server User Privilege: Administrator

Any user created with the user privilege of administrator can perform any of the following functions:

- The administrator user can be a general user when logging in to a Mobile application on a device, which is the same as described in Section 5.1.1.2, "Mobile Server User Privilege: User".
- The administrator can publish applications either through the Packaging Wizard or through the Mobile Manager.
- The administrator has authorization to use the Mobile Manager.

Once an administrator user is created, it must be associated with the Mobile Manager in the same manner that an ordinary Mobile Server user is associated with any application. See Section 5.1.1.3, "Associating Mobile Server Users With Published Applications" for more information on this process.

5.1.1.2 Mobile Server User Privilege: User

The Mobile Server user with privilege of user is created only for accessing and synchronizing published applications and its data. The user has a specific username/password for synchronizing the application from a device. Thus, this Mobile Server user enables access to a particular Mobile application and its publication items. That is, in order for the Palm, Windows CE, or other devices to be able to synchronize and retrieve a snapshot of data from the database, the Mobile Server validates that the username/password that is entered is valid for the application. If it is, then Mobile Server enables the device to retrieve the snapshot that is indicated by the publication items packaged with the application.

After creating the user, the administrator associates the user with the published applications from which this user will receive data. In addition, if any of the publication items require a parameter to be set, the administrator also sets this parameter for each user. See Section 5.1.1.3, "Associating Mobile Server Users With Published Applications" for more information.

5.1.1.3 Associating Mobile Server Users With Published Applications

Any user that wants to use an application must be associated with that application by an administrator user in the Mobile Manager. In order to associate Mobile Server users with applications, a Mobile Server administrator performs the following:

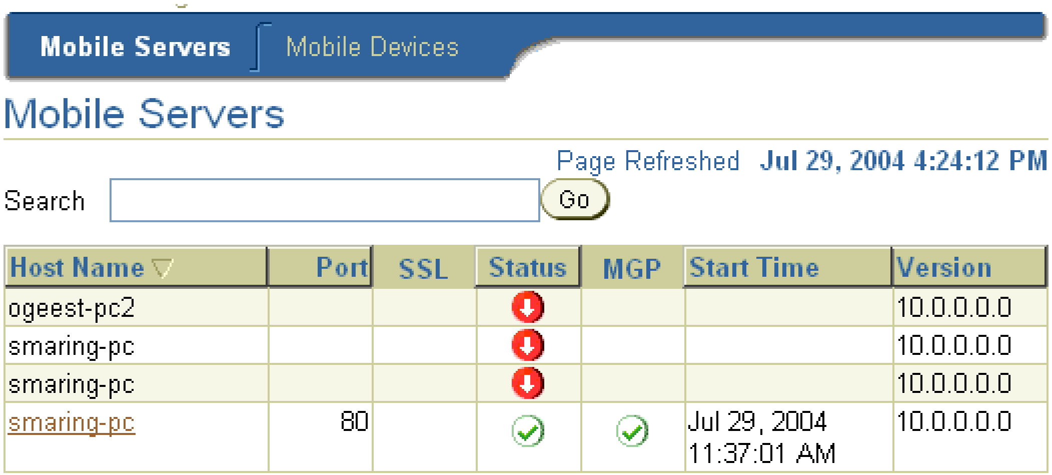
1. Package and publish an application with appropriate publication(s).
2. Create one or more users or groups that will use the application to retrieve data from the database down to a device. See Section 5.1.3, "Adding New Users" for more information.
3. Associate the users or groups with the application. See Section 5.2.1, "Grant or Revoke Application Access to Users" for more information.
4. Optionally, if the application has parameters, also known as data subsetting, that are set for each user or group, define these parameters for each user or group. See Section 5.3, "Managing Application Parameter Input (Data Subsetting)" for more information.

5.1.2 Displaying Users

You can see what users and groups have been created with all information relevant to users—such as user names and so on. If you are using OID as your repository, then the users that exist within OID are also displayed, but are not enabled for Mobile Server. You can enable these users within OID by checking the Enabled box next to the name on the Users screen.

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 Servers Farm page is displayed.

Figure 5–1 Mobile Server Farms Page



Mobile Servers

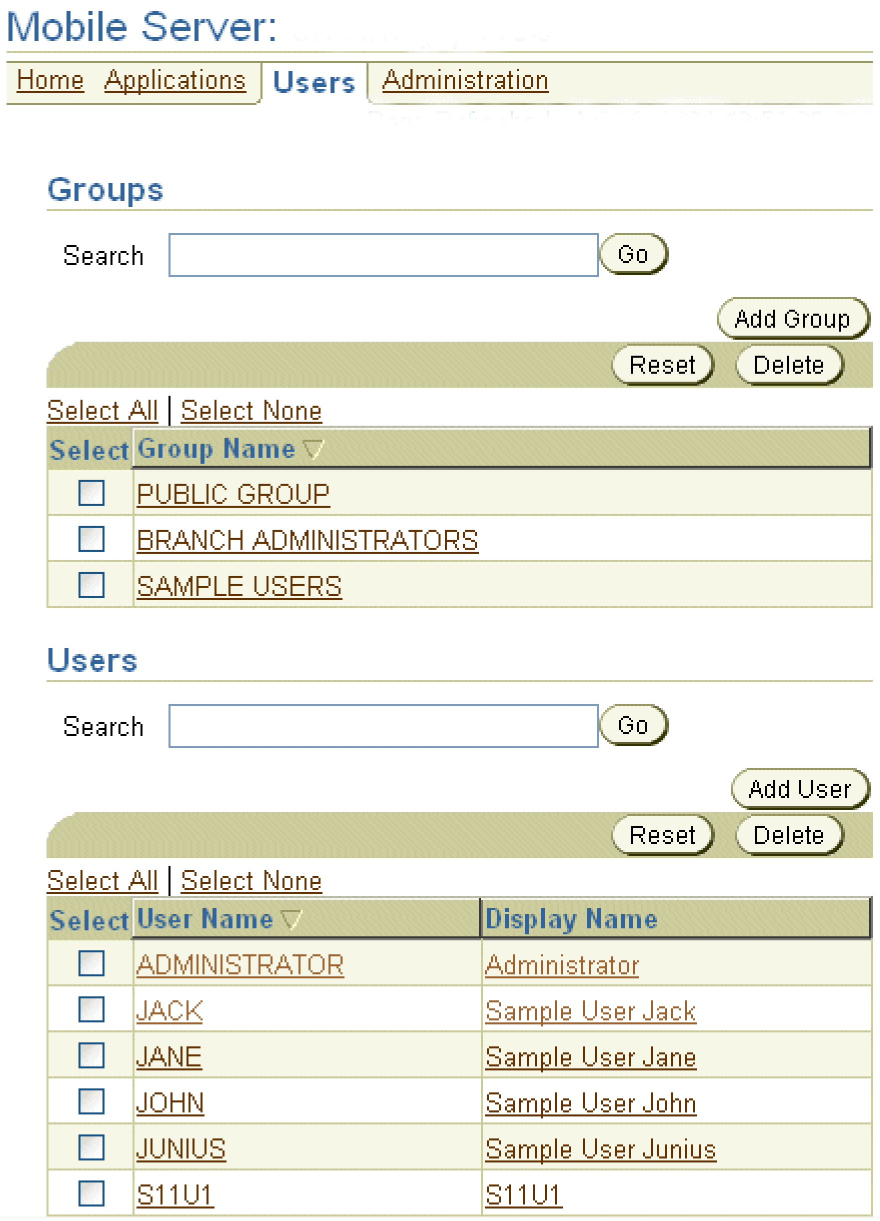
Page Refreshed **Jul 29, 2004 4:24:12 PM**

Search [ ] Go

Host Name ▽	Port	SSL	Status	MGP	Start Time	Version
ogeest-pc2			⬇			10.0.0.0.0
smaring-pc			⬇			10.0.0.0.0
smaring-pc			⬇			10.0.0.0.0
smaring-pc	80		✓	✓	Jul 29, 2004 11:37:01 AM	10.0.0.0.0

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



Select	Group Name ▽
□	PUBLIC GROUP
□	BRANCH ADMINISTRATORS
□	SAMPLE USERS

Select	User Name ▽	Display Name
□	ADMINISTRATOR	Administrator
□	JACK	Sample User Jack
□	JANE	Sample User Jane
□	JOHN	Sample User John
□	JUNIUS	Sample User Junius
□	S11U1	S11U1

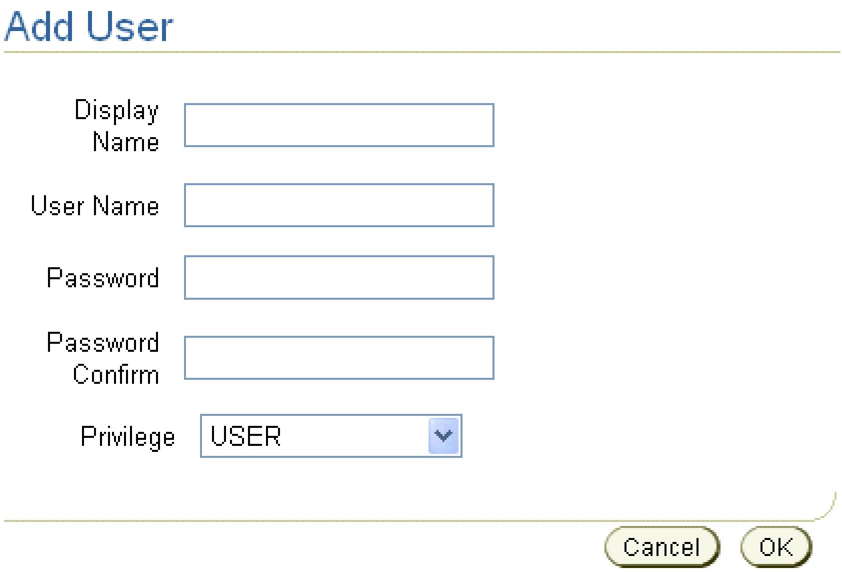
Searching Group Names or User Names

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.

5.1.3 Adding New Users

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

Figure 5–3 Add User Page



Add User

Privilege: USER

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	Optional. Password used to logon to the Mobile Server.
Password Confirm	Optional. To confirm the above mentioned password, re-enter your password.
Privilege	Lists available privileges for the Mobile Server user. ■ The **Administrator** option provides privileges to modify Mobile Server resources. ■ The **User** option provides access for registered users to the Mobile Server. For a description of each privilege type, see Section 5.1.1, "What Are Mobile Server Users?"

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 blank spaces.

Deleting Groups or Individual Users

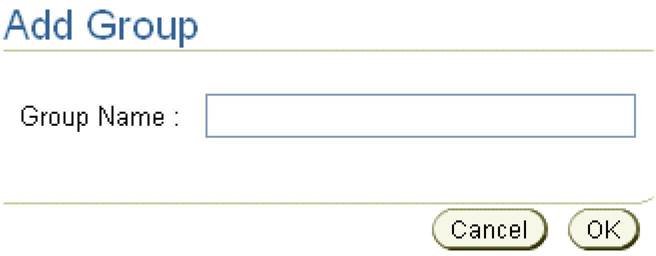
As an administrator, you can delete groups or individual users from the system. To permanently delete groups or individual users from the system, select the **Delete** check box against the group name or individual user name that you want to delete, and click **Delete**. The Mobile Manager seeks your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

5.1.4 Adding New Groups

If you have several users that require access to the same application, you can bypass adding access rights for each user by including these users in a group. Once all of the users are included in a group, then assign access to the intended application to the group; at this point, all users in the group have access to the application.

As an administrator, you can add a new group that accesses the Mobile Server. To add a new group, navigate to the Users page and click **Add Group**. As Figure 5–4 displays, the Add Group page appears and lists the requisite criteria to register user group properties.

Figure 5–4 Add Group Page



Add Group

Group Name :

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

5.1.5 Managing OID Users in the Mobile Server

If you want, you can use the Oracle Internet Directory (OID) for storing and retrieving user information instead of the Mobile Server Repository. To facilitate using OID, you must first migrate all user information from the repository into OID. Once migrated, you can use OID instead of the repository.

OID is part of the Oracle9*i*AS or OracleAS application server.

If you decide to use OID users (from OracleAS), then after you install the application server and Oracle Mobile Lite, perform the following:

1. If you currently have installed the Mobile Server and have existing users in the Mobile Server, then you must migrate any existing Mobile users to OID (See Section 3.6, "Migrate Your Users From the Mobile Server Repository to the Oracle Internet Directory" in the *Oracle Database Lite Getting Started Guide*).
2. Set the `SSO_ENABLED` parameter in the `webtogo.ora` file to `YES`. In the Mobile Manager, migrate Administration tab and select **Edit Config file**. This is the `webtogo.ora` file.
3. Restart the application server. When you modify the `SSO_ENABLED` parameter, the Mobile Server modifies the application server configuration.
4. Enable OID users for the Mobile Server.

> **Note:** When you navigate to the Users page in the Mobile Manager, all OID users are displayed. Add any new users through OID. On this page, you can only enable OID users for use within the Mobile Server or change the password.

To enable OID users for the Mobile Server, select the user and click **Enable**.

5. Assign the appropriate application to these users. As with any Mobile Server user, you must grant access to the appropriate applications. See Section 5.2.1, "Grant or Revoke Application Access to Users" for more information.

5.2 Managing Access Privileges for Users and Groups

The Mobile Server Administrator grant access privileges to Mobile applications by designating the users that can access these applications. The following sections describe the access feature of the Mobile Server. Topics include:

- Section 5.2.1, "Grant or Revoke Application Access to Users"
- Section 5.2.2, "Include or Exclude Users from Group Based Access"
- Section 5.2.3, "Grant or Revoke Application Access to Groups"

5.2.1 Grant or Revoke Application Access to Users

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

- Grant Application Access to Users
- Revoke Application Access to Users

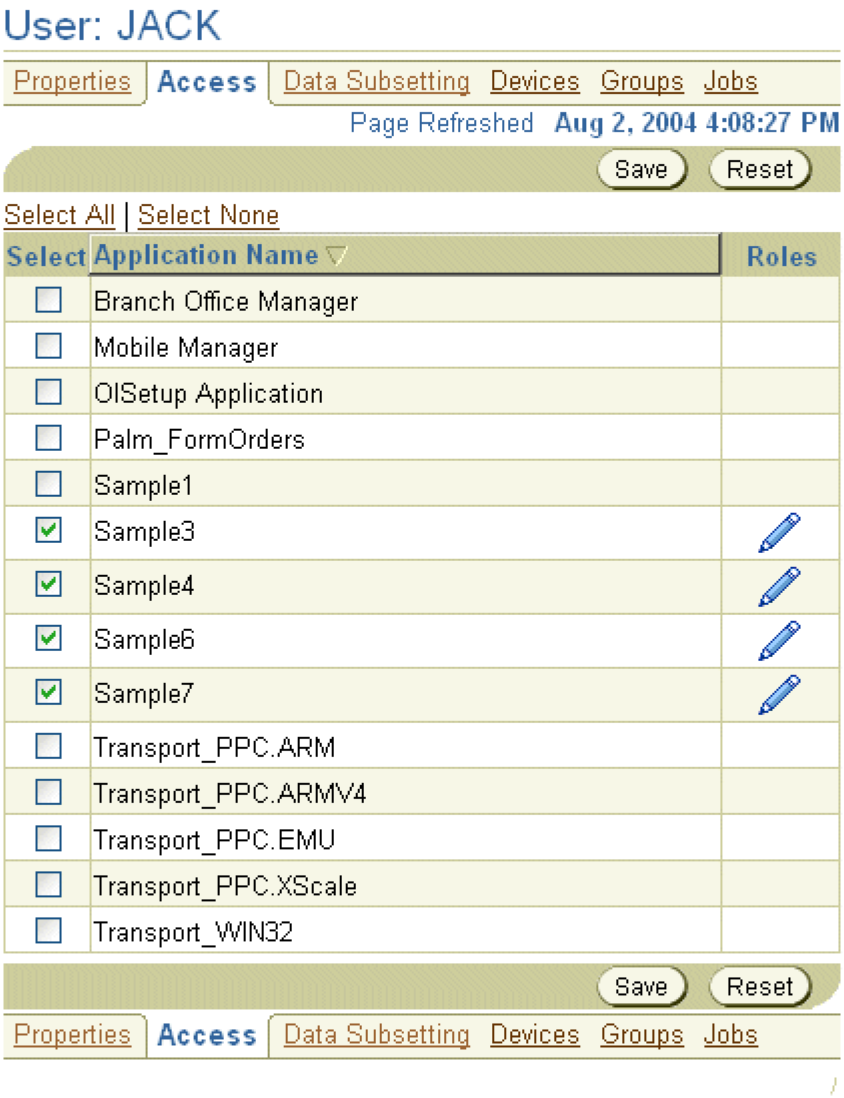
Grant Application Access to Users

The administrator can grant access to applications for specific users within the Mobile Manager, as follows:

1. Navigate to the Users page. Click the specific user name to which you wish to give access. This user's Properties page appears.
2. Click **Access**. The Access page displays a list of published applications.
3. Select the checkbox next to each application that you wish to give access to for this particular user.
4. Click **Save**.

As Figure 5–5 displays, the Access page displays a list of available applications for the user Jack. Select the applications that you want Jack to have access to and click **Save**. In this example, Jack is given access to Sample1, Sample3, Sample4, Sample6, and Sample7 applications.

Figure 5–5 Granting Application Access



Page Refreshed Aug 2, 2004 4:08:27 PM

Select	Application Name ▽	Roles
☐	Branch Office Manager	
☐	Mobile Manager	
☐	OlSetup Application	
☐	Palm_FormOrders	
☐	Sample1	
☑	Sample3	✎
☑	Sample4	✎
☑	Sample6	✎
☑	Sample7	✎
☐	Transport_PPC.ARM	
☐	Transport_PPC.ARMV4	
☐	Transport_PPC.EMU	
☐	Transport_PPC.XScale	
☐	Transport_WIN32	

Revoke Application Access to Users

To revoke application access to any user, 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. For directions on how to include or exclude any user from a group, see Section 5.2.2, "Include or Exclude Users from Group Based Access".

5.2.2 Include or Exclude Users from Group Based Access

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

- Include Users in a Group
- Exclude Users from a Group

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.

Include Users in a Group

To include users into a group, do the following:

1. Navigate to the Users page. Click on the username of the user you wish to include in a group. This user's Properties page appears.
2. Click **Groups**.
3. Select the group name that you want to include the user into.
4. Click **Save**.

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

Now the user takes on the access for all applications to which the group has access. In order for the group to be given access to additional applications, follow the instructions in Section 5.2.1, "Grant or Revoke Application Access to Users". However, instead of selecting a particular user, select the group instead.

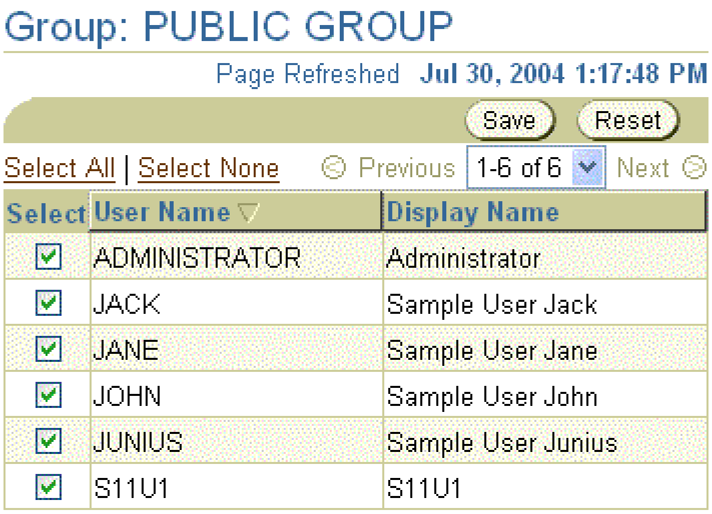
Exclude Users from a Group

To remove a user from any group, do the following:

1. Navigate to the Users page. Click on the username of the user you wish to exclude from a group. This user's Properties page appears.
2. Click **Groups**.
3. Clear the group name that you want to exclude the user from.
4. Click **Save**.

Figure 5–6 displays the Clear Group page for the Public Group. If you wanted to clear Jack from this group, you would uncheck the checkbox next to Jack's name and click Save.

Figure 5–6 Clear Group Page



Group: PUBLIC GROUP

Page Refreshed Jul 30, 2004 1:17:48 PM

Select	User Name ▽	Display Name
☑	ADMINISTRATOR	Administrator
☑	JACK	Sample User Jack
☑	JANE	Sample User Jane
☑	JOHN	Sample User John
☑	JUNIUS	Sample User Junius
☑	S11U1	S11U1

5.2.3 Grant or Revoke Application Access to Groups

Once you have the users that you want in a group, you must indicate what applications that the group has access to. In order to assign application access to groups, you have to add the access rights off the application page. See Section 4.5.1, "Grant Application Access to Users and Groups" for directions.

5.3 Managing Application Parameter Input (Data Subsetting)

If the application that this user accesses requires one or more parameters to determine what data is retrieved from the database, you set these parameters, also known as data subsetting, within the user configuration in Mobile Manager.

> **Note:** You can only set the parameter values once a user has been granted access to the application. See Section 5.2, "Managing Access Privileges for Users and Groups" for instructions.

For example, if you have an application that retrieves the customer base for each sales manager, the application needs to know the sales manager's identification number to retrieve the data specific to each manager. The identification number, in this example, is the application parameter required that is associated with this user. Thus, if you set up each sales manager as a unique user and set their identification number in the data subsetting screen, then the application is given that unique information and can replace it appropriately in the application.

1. Navigate to the Users page. Click the specific user name to which you wish to give access. This user's Properties page appears.
2. Click **Data Subsetting**. The Data Subsetting page enables the administrator to add parameter input for this user. This displays all of the applications that the user is associated with.
3. Select the application for which you want to add the parameter value.
4. Enter the parameter values for the application.
5. Click **Save**.

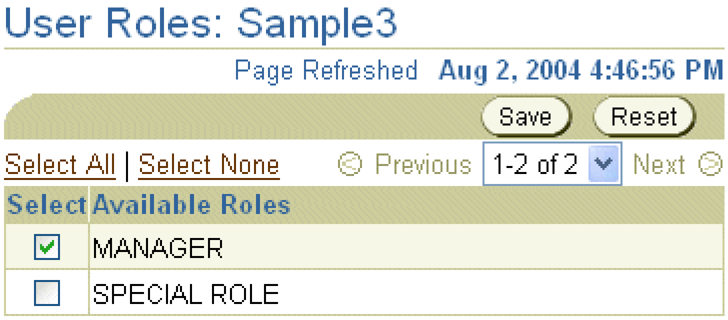
5.4 Assigning Application Roles to Users

When the developers design any Web-to-Go application, they can include functionality that is enabled based on the role that the user is assigned. For example, if you have a manager and employee role in an application, the user who is assigned the manager role may have other options available to view on the application GUI. These options would not show up for those users who are assigned the employee role. See Section 4.2.2, "Application Roles" and Section 2.5.5, "Granting Roles" in the *Oracle Database Lite Developer's Guide* for information on how to programmatically create and grant these roles.

Once the application is deployed, all roles are displayed and can be assigned to any user in the Mobile Manager. You can assign roles either through the Mobile Manager or through the `wsh` script. This section describes how to assign users to certain roles for a Web-to-Go application.

Figure 5–5 displays the User page for Jack. Notice that there is a column for Roles. If you click the pencil icon in this column, you can see the roles that have been created in the application. For example, if we click on the pencil icon for the Sample3 application, as shown in Figure 5–7, we see that two roles have been created in this application: Manager and Special Role. Select the checkbox next to any of the roles to which you want Jack to be added. In this case, the Manager role is checked, so Jack will be added to the Manager role.

Figure 5–7 Add Jack to the Sample3 Application Manager Role



User Roles: Sample3

Page Refreshed Aug 2, 2004 4:46:56 PM

Select	Available Roles
☑	MANAGER
□	SPECIAL ROLE

5.5 Creating an Administrator

As referenced in the previous sections, to create any user, including administrators, you must do the following:

1. Create one or more users or groups that will use the application to retrieve data from the database down to a device. See Section 5.1.3, "Adding New Users" for more information.

2. Associate the users or groups with the application. See Section 5.2.1, "Grant or Revoke Application Access to Users" for more information.

3. Optionally, if the application has a parameter, also known as data subsetting, that is set for each user or group, define the parameters for each user or group. See Section 5.3, "Managing Application Parameter Input (Data Subsetting)" for more information.

Thus, to create an administrator, you would do the following:

1. Create a user with the name of the administrator that you want, with the privilege of administrator.

2. Navigate to the Access tab for this new administrator and check the checkbox next to Mobile Manager.

You now have a new administrator user. You can log into your Mobile Manager with this user's name and password.

5.6 Manually Adding Devices for a User

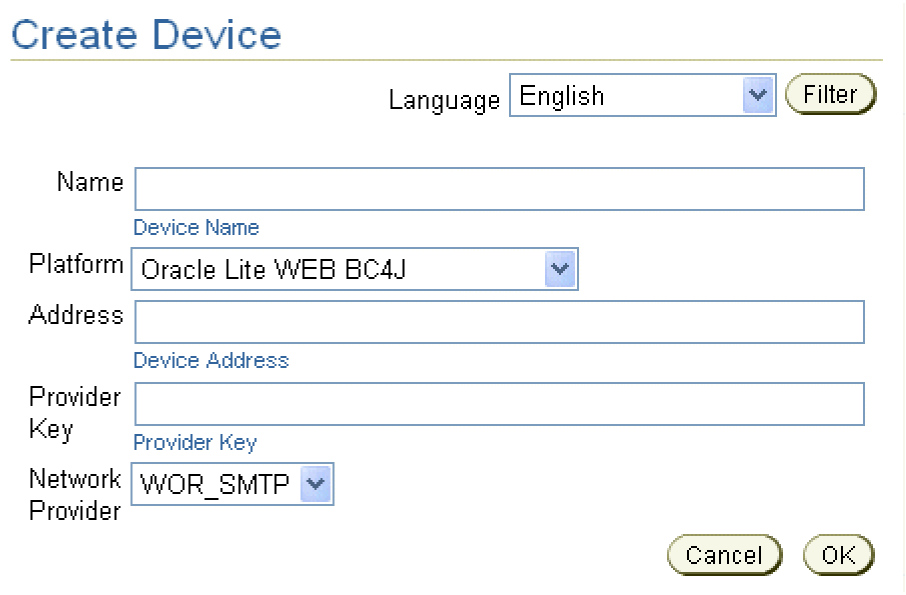
Normally, when you download and install a client, the device is registered automatically for the user. There are two instances where you may need to manually add the device:

- As an administrator, you could hand a device that is fully loaded with the Mobile client software, but is not assigned to any user or application. After handing the device to your user, you can add their user information, application access, and device that they are using manually.
- When you hand someone the Mobile client software on an installation CD, then the installation does not register the device manually—since it is not connected to Mobile Server. Thus, for each user that you provide the Mobile client software from an install CD, you will have to add the device to this user.

To add a device for an individual user, navigate to the specific user's page and do the following:

1. On the Users page, select the user for which you want to add a device.
2. Click **Devices**. All currently registered devices for this user appear.
3. Click **Add**. The Create Device screen (as shown in Figure 5–8) appears.

Figure 5–8 Manually Add Device to User



Create Device

Language: English

Name:

Platform: Oracle Lite WEB BC4J

Address:

Provider Key:

Network Provider: WOR_SMTP

4. Enter the device information, as described in Figure 5–2, and click **OK** to add the device for this user:

Table 5–2 Device Information

Device Field	Description
Language	Select the language that the platform will use. The default is English.
Name	Configure a user-defined name for the device.
Platform	Select the platform for this device.
Address	
Proicer Key	
Network Provider	

Once added, the user can now synchronize the device to retrieve their applications and related snapshots.

5.7 Set Update Policy for Software Updates for the User

You can control whether a new version of an application software is downloaded on each client. Modify the update policy attribute of the user with the Software Update pulldown to the appropriate update that you want, as follows:

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

In addition, you can specify the date that the update occurs.

6 Managing Synchronization

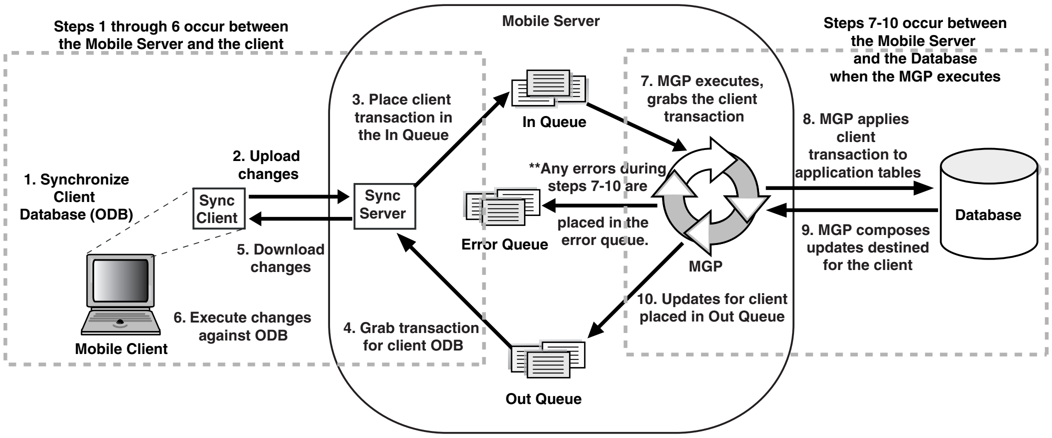
The Mobile Server administrator uses the Data Synchronization Manager to manage synchronization tasks. This chapter includes:

6.1 How Does the Synchronization Process Work?

The Mobile Server uses synchronization to replicate data between the Mobile clients with their client Oracle Lite databases (including those for Web-to-Go, Win32, Palm, and Windows CE platforms) and the application tables, which are stored on a back-end Oracle database.

When most people think of synchronizing data, they think of their Palm Pilot. When you hit the synchronization button for the Palm Pilot, any changes are added to the database of information on the Windows machine immediately. This is not the case for Oracle Database Lite, which is used for multiple clients. In order to accomodate multiple users, the application tables on the back-end database cannot be locked by a single user. Thus, the synchronization process involves using queues to manage the information between the Mobile clients and the application tables in the database (demonstrated by Figure 6–1), as follows:

Figure 6–1 Data Synchronization Architecture


Steps 1 through 6 occur between the Mobile Server and the client
Mobile Server
Steps 7-10 occur between the Mobile Server and the Database when the MGP executes
3. Place client transaction in the In Queue
In Queue
7. MGP executes, grabs the client transaction
8. MGP applies client transaction to application tables
2. Upload changes
1. Synchronize Client Database (ODB)
Sync Client
Sync Server
**Any errors during steps 7-10 are placed in the error queue.
Error Queue
Database
9. MGP composes updates destined for the client
MGP
5. Download changes
10. Updates for client placed in Out Queue
6. Execute changes against ODB
4. Grab transaction for client ODB
Mobile Client
Out Queue

1. User initiates a synchronization from the Mobile client. Note that the Mobile client may be a Windows platform client or a PDA.
2. Mobile client software gathers all of the client changes into a transaction and the Sync Client uploads the transaction to the Sync Server on the Mobile Server.
3. Sync Server places the transaction into the In-Queue.

> **Note:** When packaging your application, you can specify if the transaction is to be applied at the same time as the synchronization. If you set this option, then the transaction is immediately applied to the application tables. However, note that this may not be scaleable and you should only do this if the application of the transaction immediately is important and you have enough resources to handle the load.

4. Sync Server gathers all transactions destined for the Mobile client from the Out-Queue.
5. Sync Server transfers these transactions down to the Sync Client.
6. Mobile client downloads and applies all changes for client Oracle Lite database.
7. All transactions compiled from all Mobile clients are gathered by the MGP out of the In-Queue.
8. The MGP applies all transactions for the Mobile clients to their respective application tables.
9. Any updates destined for any Mobile client is composed into a transaction by the the MGP process.
10. MGP places outgoing transactions for Mobile clients into the Out-Queue, waiting for the next client synchronization for the Sync Server to gather the updates to the client.

As Figure 6–1 demonstrates, synchronization is broken up into two phases: the Mobile client using the Sync Client/Server to upload and download changes and the MGP process interacting with the queues and the application tables to apply and compose transactions. These phases are displayed separately in the Data Synchronization section of the Mobile Manager.

On the Mobile Server home page, you can navigate to the Data Synchronization home page by clicking Data Synchronization, which is located under the Components section.

6.2 Managing the Sync Server

The Sync Server is an HTTP servlet that listens to client synchronization requests. As demonstrated by Figure 6–1, during every synchronization session, the Sync Server uploads client transactions from the client Oracle Lite database and places them within the In-Queues. The Sync Server then downloads any server-side transactions from the Out-Queues to the client Oracle Lite database.

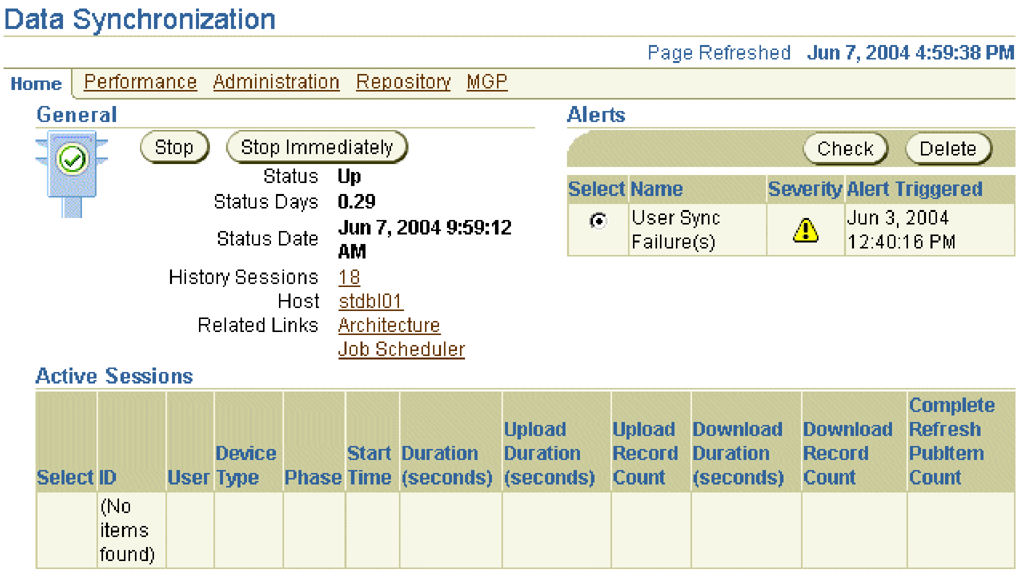
From the Data Synchronization home page, you can manage Sync Server tasks—such as the following:

- Section 6.2.1, "Starting/Stopping the Sync Server"
- Section 6.2.2, "Checking Synchronization Alerts"
- Section 6.2.3, "Managing Sync Sessions"
- Section 6.2.4, "Displaying Operating System (OS) and Java Virtual Machine (JVM) Information"

6.2.1 Starting/Stopping the Sync Server

To start the Sync Server, navigate to the Data Synchronization home page. The Sync Server default status is Up, as displayed in Figure 6–2.

Figure 6–2 Data Synchronization Home Page



Data Synchronization

Page Refreshed Jun 7, 2004 4:59:38 PM

General

Status	Up
Status Days	0.29
Status Date	Jun 7, 2004 9:59:12 AM
History Sessions	18
Host	stdbl01
Related Links	Architecture Job Scheduler

Alerts

Select	Name	Severity	Alert Triggered
●	User Sync Failure(s)	⚠	Jun 3, 2004 12:40:16 PM

Active Sessions

Select	ID	User	Device Type	Phase	Start Time	Duration (seconds)	Upload Duration (seconds)	Upload Record Count	Download Duration (seconds)	Download Record Count	Complete Refresh PubItem Count
	(No items found)										

To gracefully shut down the Sync Server, click **Stop**. The Sync Server stops after all current sessions have completed synchronization. To immediately stop the Sync Server, click **Stop Immediately**, which kills current sync sessions immediately. Use for emergency situations.

6.2.2 Checking Synchronization Alerts

Both the Sync Server and MGP register alerts if a problem occurs within any part of the synchronization phases. There are two types of alerts, as follows:

- Critical alerts—For the Sync Server, clients cannot synchronize if the Sync Server encounters an exception (also known as a critical alert); thus, the errors must be resolved by the administrator. Once resolved, the administrator re-starts the Sync Server.
- Warning alerts—These alerts are registered when an individual synchronization session fails. The administrator checks the Sync session details in the Sync Session History and determines the reasons for the failure. If necessary, the administrator may need to involve a DBA, if the reason is database-related.

Each alert provides the alert name, degree of severity, time when the alert was triggered, and time when the alert was last checked by a DBA.

Table 6–1 lists sample alerts. Note that the type designates whether the alert originates from the Sync Server or the MGP.

Table 6–1 Alert Types

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

6.2.3 Managing Sync Sessions

For all users, the sessions that are currently in the process of a synchronization are displayed in the Active Sessions table at the bottom of the Data Synchronization Home Page. Synchronization involves uploading or downloading updates between the the Sync Client and Sync Server.

You can terminate any active session on the Data Synchronization Home page by performing the following steps:

1. Select the active session that you wish to terminate and click **Kill**.
2. Click **Yes**.
3. Click **OK** for the confirmation message.

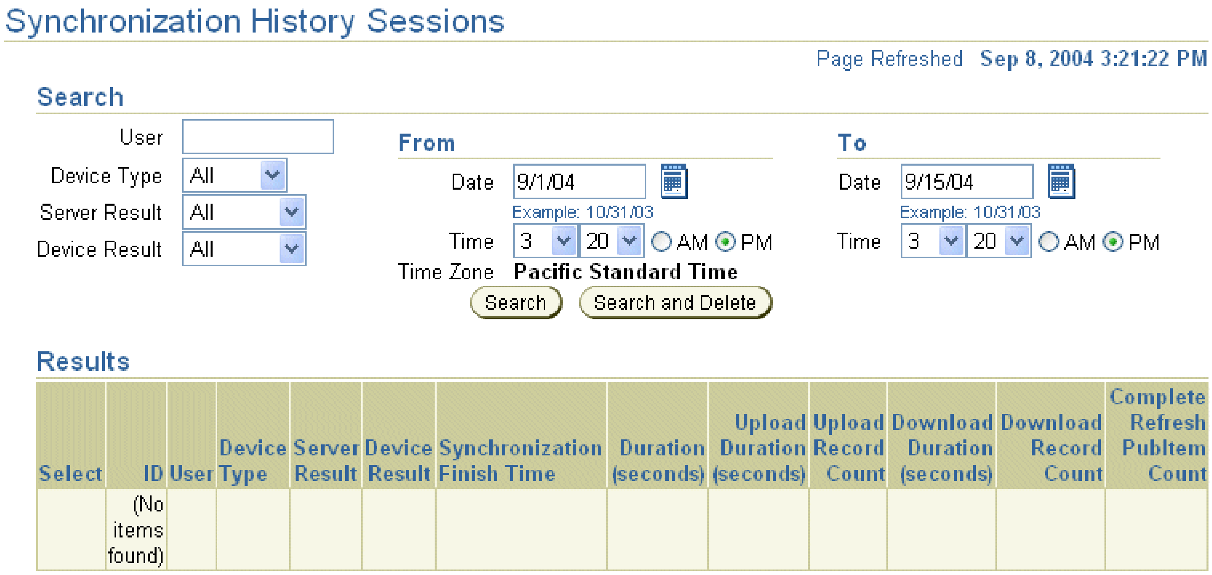
The Active Sessions table on the Data Synchronization home page also displays session details. Select the active session that you wish to view and click **Details** to see the publication items that have been uploaded or downloaded, waiting publication items, records and timing information, and the session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

If you want to view all details about completed synchronization sessions, navigate to the Synchronization History Sessions screen. To navigate to this screen, either click the number hyperlink next to History Sessions on the home page or navigate through the Performance tab. The total number of registered sessions is designated by the number next to History Sessions.

Note: The session history for each user between the Sync Client and Sync Server is saved only if you set the `SYNC_HISTORY` parameter to `YES`, which is the default. You can set the `SYNC_HISTORY` instance parameter to `YES` or `NO` by navigating to `Data Synchronization->Administration->Instance Parameters`.

Figure 6–3 shows the Synchronization History Sessions page.

Figure 6–3 Synchronization History Sessions Page



Synchronization History Sessions

Page Refreshed Sep 8, 2004 3:21:22 PM

Search

User

Device Type: All

Server Result: All

Device Result: All

From

Date: 9/1/04

Example: 10/31/03

Time: 3 : 20 ○ AM ● PM

Time Zone: Pacific Standard Time

To

Date: 9/15/04

Example: 10/31/03

Time: 3 : 20 ○ AM ● PM

Results

Select	ID	User	Device Type	Server Result	Device Result	Synchronization Finish Time	Duration (seconds)	Upload Duration (seconds)	Upload Record Count	Download Duration (seconds)	Download Record Count	Complete Refresh PublItem Count
	(No items found)											

All session history is not displayed until you search for the appropriate records. If you want all records within a specified date, then the only thing that you need to provide is the From and To date range and click **Search**. Be careful to only click **Search and Delete** if you want these records removed. You can further narrow the search by specifying one or more of the following:

- The name of the user from which all synchronizations originated
- The device platform type to see all synchronizations from just these platforms.
- Only those synchronizations that were successful or failures from the server-side.
- Only those synchronizations that were successful or failures from the device-side.

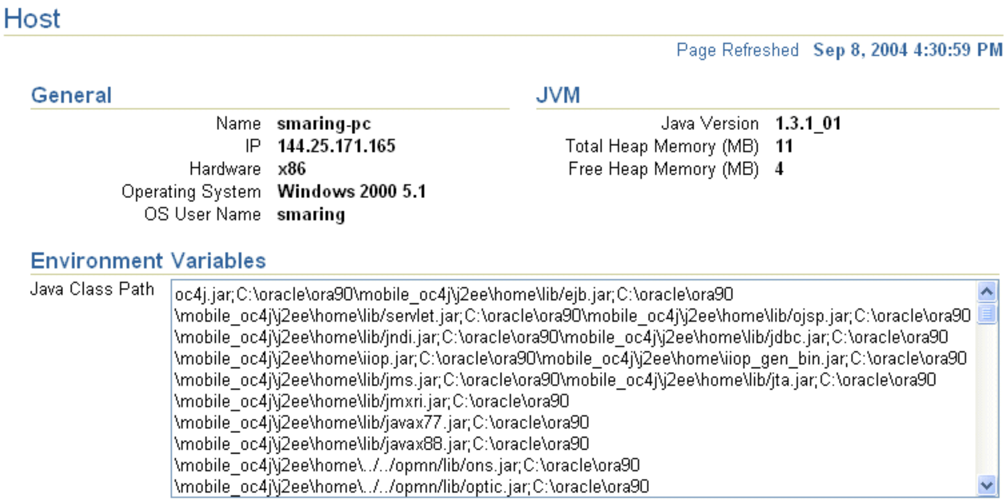
The Session History page displays matched sessions in the **Results** section. Once displayed, you can sort by most of the headers to either sort top to bottom or bottom to top. For example, to sort sync sessions by user, click the **User** header title.

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.

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

You can see the operating system and JVM versions that are installed on the host where the Mobile Server resides by clicking the **Host** hyperlink that is displayed against the Host on the Data Synchronization home page. As displayed in Figure 6–4, 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 6–4 Host Page



Host

Page Refreshed **Sep 8, 2004 4:30:59 PM**

General

Name	**smaring-pc**
IP	**144.25.171.165**
Hardware	**x86**
Operating System	**Windows 2000 5.1**
OS User Name	**smaring**

JVM

Java Version	**1.3.1_01**
Total Heap Memory (MB)	**11**
Free Heap Memory (MB)	**4**

Environment Variables

Java Class Path

```
oc4j.jar;C:\oracle\ora90\mobile_oc4j\j2ee\home\lib/ejb.jar;C:\oracle\ora90
\mobile_oc4j\j2ee\home\lib/servlet.jar;C:\oracle\ora90\mobile_oc4j\j2ee\home\lib/ojsp.jar;C:\oracle\ora90
\mobile_oc4j\j2ee\home\lib/jndi.jar;C:\oracle\ora90\mobile_oc4j\j2ee\home\lib/jdbc.jar;C:\oracle\ora90
\mobile_oc4j\j2ee\home\iiop.jar;C:\oracle\ora90\mobile_oc4j\j2ee\home\iiop_gen_bin.jar;C:\oracle\ora90
\mobile_oc4j\j2ee\home\lib/jms.jar;C:\oracle\ora90\mobile_oc4j\j2ee\home\lib/jta.jar;C:\oracle\ora90
\mobile_oc4j\j2ee\home\lib/jmxri.jar;C:\oracle\ora90
\mobile_oc4j\j2ee\home\lib/javax77.jar;C:\oracle\ora90
\mobile_oc4j\j2ee\home\lib/javax88.jar;C:\oracle\ora90
\mobile_oc4j\j2ee\home\../../opmn/lib/ons.jar;C:\oracle\ora90
\mobile_oc4j\j2ee\home\../../opmn/lib/optic.jar;C:\oracle\ora90
```

6.3 Configuring Data Synchronization

There are two types of configuration parameters for Data Synchronization:

- Shared—Shared parameters affect all Mobile Server instances in the farm. The administrator can have multiple Mobile Server instances in a single farm that uses the same Mobile repository. To modify these parameters, navigate to the Administration screen and click **Shared Parameters**.
- Instance—Instance parameters only affect a single Mobile Server instance; that is, the Mobile Server that you are currently viewing. These parameters are stored in the `WEBTOGO.ORA` file; thus, once modified, you may need to restart the Mobile Server. Check the Need Restart column to verify if a restart is necessary. To modify these parameters, navigate to the Administration screen and click **Instance Parameters**. See Appendix B, "Configuration Parameters for the WEBTOGO.ORA File" for a description of each of these parameters.

  It is never recommended to modify the `webtogo.ora` file directly; instead, use the Mobile Manager to modify any of the `webtogo.ora` file parameters.

To view the parameter description and additional information, click **Show**. You can modify any of the values in the **New Value** field and click **Apply**. Some of the Instance parameter values do not take effect until the Mobile Server is restarted.

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

In the default server configuration, Mobile clients do not automatically encrypt the snapshot ODB files after you complete the initial sync. However, you can modify your configuration to automatically encrypt all snapshot ODB files with the synchronization user password after the initial sync completes. The following sections demonstrate how to configure for this option either on the client or on the server:

- Section 6.4.1, "Configuring on the Local Client for Automatic Encryption of Local Snapshots"
- Section 6.4.2, "Configuring on the Server for Automatic Encryption of Local Snapshots"

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

On the local client, you can configure for automatic encryption of the snapshot ODB files after initial synchronization by modifying the POLITE.INI/POLITE.TXT file with the following parameter:

```
[SYNC]
ENCRYPT_DB=1
```

After the first encryption, you may want to modify the parameter to a 2, to eliminate the encryption (and performance issue) after every synchronization. For more information on modifying the ENCRYPT_DB parameter in the POLITE.INI/POLITE.TXT file, see Section H.3.2.13, "ENCRYPT_DB".

6.4.2 Configuring on the Server for Automatic Encryption of Local Snapshots

On the server, you can configure for automatic encryption of the snapshot ODB files after initial synchronization by performing the following:

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

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

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

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

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

For more information on sending commands to the Mobile device, see Section 8.6, "Sending Commands to Your Mobile Devices".

6.5 Managing Trace Settings and Trace Files

You can configure the type of tracing that occurs for Data Synchronization components. For more information, see Section 16.1.2, "Data Synchronization Tracing".

6.6 Browsing the Repository for Synchronization Details

The Repository screen describes how to look up user information, publications, publication items, and the In-Queue, Out-Queue, and Error queues that facilitate synchronization. This section contains the following topics:

- Section 6.6.1, "Viewing User Information"
- Section 6.6.2, "Viewing Publications"
- Section 6.6.3, "Viewing Publication Items"
- Section 6.6.4, "Viewing Synchronization Queues"

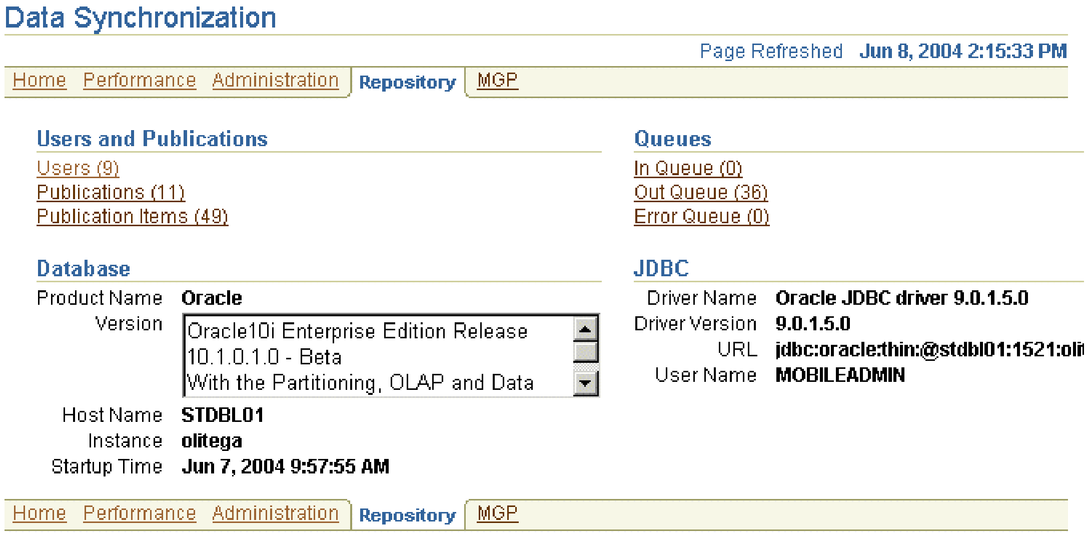
6.6.1 Viewing User Information

All users that have been added by the administrator (see Section 5.1, "Managing Users and Groups") are contained within the Mobile repository. With this Users screen, you can view everything that is attached to this user, such as application subscriptions, publication items, parameters, SQL scripts, Java resources, sequences, and performance analysis.

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

   As displayed in Figure 6–5, the Repository tab appears.

Figure 6–5 Repository Tab



Data Synchronization

Page Refreshed Jun 8, 2004 2:15:33 PM

Users and Publications

Users (9)
Publications (11)
Publication Items (49)

Queues

In Queue (0)
Out Queue (36)
Error Queue (0)

Database

Product Name Oracle
Version Oracle10i Enterprise Edition Release 10.1.0.1.0 - Beta With the Partitioning, OLAP and Data
Host Name STDBL01
Instance olitega
Startup Time Jun 7, 2004 9:57:55 AM

JDBC

Driver Name Oracle JDBC driver 9.0.1.5.0
Driver Version 9.0.1.5.0
URL jdbc:oracle:thin:@stdbl01:1521:oli
User Name MOBILEADMIN

2. Click **Users**, which brings up a list of all users currently in the repository. The number next to Users details the number of users currently in the repository.

3. Choose the user in which you are interested and click **Subscriptions**. The Subscriptions page displays the existing publications for the user. A subscription is the combination of the publication, its publications items, and the user to which it is attached.

   On the subscriptions screen, choose any publication and then click any of the buttons above it to see all of the publication items, parameters, SQL scripts, Java resources, and sequences. In addition, if you click the Consperf performance analysis button, you can generate performance analysis for the publication items.

See Section 6.7.3, "Analyzing Performance of Publications With the Consperf Utility" for more information on Consperf performance analysis.

You can add subscriptions to the user by granting the user access to the application that contains the publication. See Section 5.2.1, "Grant or Revoke Application Access to Users" on how to grant access to applications. To add a publication to an application, use the Mobile Development Workbench.

6.6.2 Viewing Publications

To view all publications that have been published against the Mobile Server, click **Publications** under the Users and Publications section. The number next to Publications are the number of publications currently in the repository, which were uploaded to the repository when the application was published. You can view these publications individually using this link. Clicking **Publications** brings up a screen that contains a list of all of the publications. If there are too many to fit on a page, you can search for a specific publication. Similar to the Users screen, you can select a publication and then view the publication items, parameters, SQL scripts, Java resources, sequences, and users that are attached to this publication.

When you add publication items to each publication, you specify certain properties for each publication item within the publication, such as the order weight of when this item is executed in relation to the other publication items in the subscription, who wins when a conflict occurs, and options for disabling DML. You can view some of these properties when you select the publication and click **Publication Items**. For more information on these properties, see Section 3.5.9 "Adding Publication Items to a Publication" in the *Oracle Database Lite Developer's Guide*.

You can only view publications in this screen. To modify your publication, use the Mobile Development Workbench. For more information, see the MDW chapter in the *Oracle Database Lite Developer's Guide*.

6.6.3 Viewing Publication Items

To view publication items, click **Publication Items** under the Users and Publications section. The number next to Publication Items details the number of publication items stored in the repository. These items were uploaded to the repository when the application was published.

Click **Show** to view the publication item properties.

You can only view publication items in this screen. To modify you publication and its publication item, use the Mobile Development Workbench. For more information, see the MDW chapter in the *Oracle Database Lite Developer's Guide*.

6.6.4 Viewing Synchronization Queues

You can view what is currently in the synchronization 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**. The number next to each queue shows the number of transactions contained within that queue.

The In-Queue and Error Queue are organized by transactions.

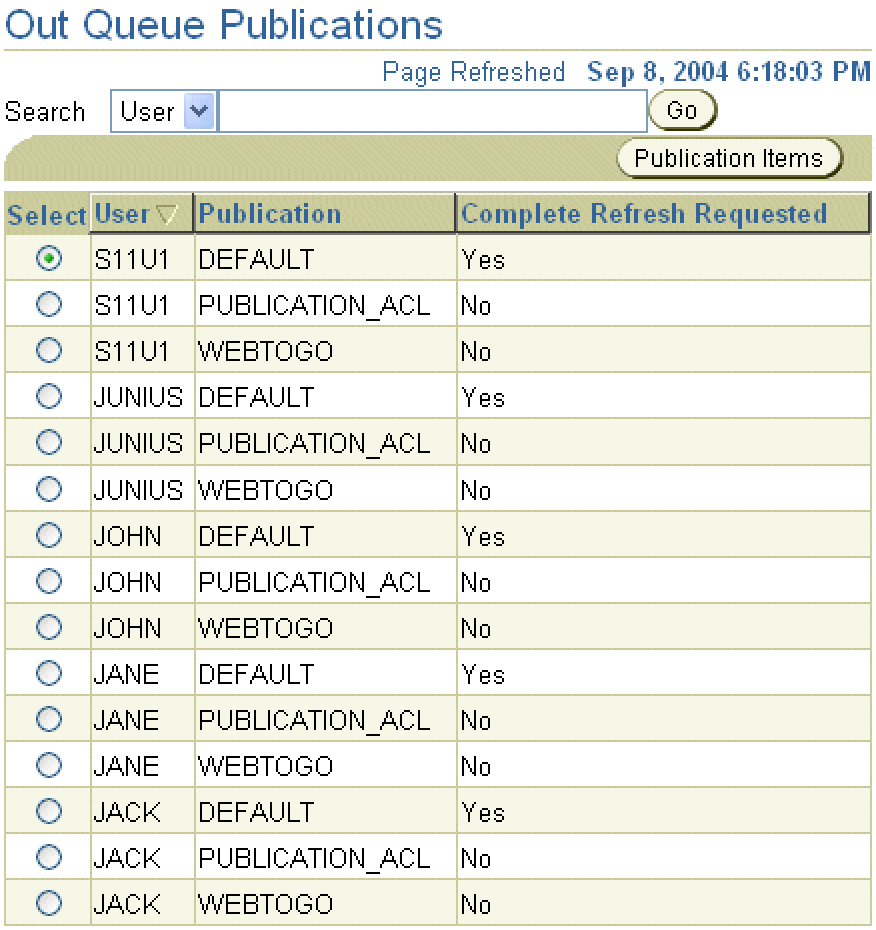
6.6.4.1 Viewing Transactions in the In-Queue

You can view the current transactions that exist in the In-Queue. If you are wondering if your changes have been applied to the application tables, you can verify if they are still in the In-Queue or have already been processed by the MGP. If you see your transactions held in the In-Queue longer than you wish, then modify the timing on how often the MGP executes in the Job Scheduler. See Section 7.3, "Manage Scheduled Jobs Using the Mobile Manager" for more information on the Job Scheduler.

6.6.4.2 Viewing Subscriptions in the Out-Queue

The Out-Queue contains the transactions that are destined for the Mobile client. The transactions are organized by subscriptions, which is a combination of the user and each publication for the user. Also, you can see if a complete refresh is requested. Figure 6–6 displays the Out-Queue Publications page.

Figure 6–6 Out-Queue Publications Page



Out Queue Publications

Page Refreshed Sep 8, 2004 6:18:03 PM

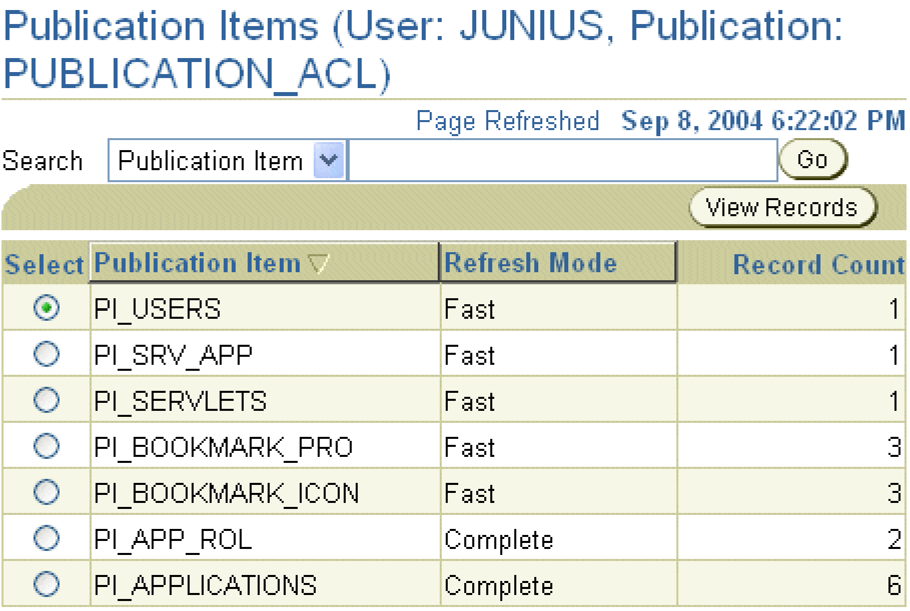
Select	User	Publication	Complete Refresh Requested
●	S11U1	DEFAULT	Yes
○	S11U1	PUBLICATION_ACL	No
○	S11U1	WEBTOGO	No
○	JUNIUS	DEFAULT	Yes
○	JUNIUS	PUBLICATION_ACL	No
○	JUNIUS	WEBTOGO	No
○	JOHN	DEFAULT	Yes
○	JOHN	PUBLICATION_ACL	No
○	JOHN	WEBTOGO	No
○	JANE	DEFAULT	Yes
○	JANE	PUBLICATION_ACL	No
○	JANE	WEBTOGO	No
○	JACK	DEFAULT	Yes
○	JACK	PUBLICATION_ACL	No
○	JACK	WEBTOGO	No

You can view the details of each subscription by performing the following:

1. Select the subscription to view with the Select button next to the user name/publication in which you are interested.
2. Click **Publication Items**, which brings up Figure 6–6.

   The Publications Items screen describes how many records is in the publication and whether it uses a fast or complete refresh mode.

Figure 6–7 Publication Items in the Out-Queue Subscription



Publication Items (User: JUNIUS, Publication: PUBLICATION_ACL)

Page Refreshed Sep 8, 2004 6:22:02 PM

Search: Publication Item

Select	Publication Item ▽	Refresh Mode	Record Count
●	PI_USERS	Fast	1
○	PI_SRV_APP	Fast	1
○	PI_SERVLETS	Fast	1
○	PI_BOOKMARK_PRO	Fast	3
○	PI_BOOKMARK_ICON	Fast	3
○	PI_APP_ROL	Complete	2
○	PI_APPLICATIONS	Complete	6

3. View the records of the publication item by clicking the Select button and then click **View Records**.
4. Click **Show** on each record to see the record data.

6.6.4.3 Viewing Transactions in the Error Queue

The purpose of the error queue is to store transactions that fail due to conflicts and other unforseen problems—such as database issues. If there is an error on the apply phase of the synchronization process, then the error is posted to this error queue. Some of the transactions placed into the error queue show unresolved error conditions, where the administrator must perform a function to resolve the problem before re-executing the transaction. These conflicts are defined with `ERROR`. However, other conflicts are resolved by the Mobile Server, such as by the conflict resolution rules of "server wins" or "client wins". With these, a transaction is still logged to the error queue to inform the administrator of how the conflict was resolved. It is up to the administrator to modify the outcome of the conflict. These messages are defined with `CONFLICT DETECTED`.

If the administrator resolves the error condition that caused the problem, then the administrator may attempt to re-apply the transaction or purge the error queues. To view the Mobile Server error queues in the database, the error queue is `C$EQ` and the data is stored in `CEQ$<base_table_name>`.

A Mobile Server synchronization conflict occurs if:

- The client and the server update the same row. This error is resolved by the Mobile Server by the conflict rules, but is logged in the error queue for you to see the result. You can choose to modify the result.
- The client and server create rows with the same primary key values. This error is resolved by the Mobile Server, but is logged in the error queue for you to see the result. You can choose to modify the result.

- The client deletes the same row that the server updates. This error is resolved by the Mobile Server, but is logged in the error queue for you to see the result. You can choose to modify the result.
- The server deletes the same row that the client updates. This error is unresolved by the Mobile Server. The administrator must decide how this is resolved.
- Client is out of sync. This error is unresolved by the Mobile Server. The administrator must decide how this is resolved.
- Client records violate server database constraints. This error is unresolved by the Mobile Server. The administrator must decide how this is resolved by either modifying the database constraints and re-executing the transaction, or by modifying how the client is able to update records to conform to the constraints.
- An error occurs when reapplying a backup. See Section 15.3, "Oracle Database Lite Backup Coordination Between Client and Server" for instructions on recovering from a backup.
- An error occurs with the back-end database, such as a constraint violation or storage issue.

If you decide to reapply the records in the transaction to the application tables, you must perform the following:

1. Correct the reason why the error occurred in the first place.
2. You can only re-execute the transaction if the DML operation is Insert, Update, or Delete. If the DML operation is in the Error state, then modify the DML operation from Error to Update. The DBA must modify the record in the error queue for the base table, named `CEQ$<base_table_name>`, changing the DML operation from Error (`E`) to Update (`U`), Insert (`I`) or Delete (`D`).
3. If the conflict resolution is set to "server wins," then you may lose the client modifications. Thus, if you set the conflict resolution to "client wins," then you force these changes to overwrite the server.
4. Once the DML operation is Insert, Update, or Delete, re-execute the command.
   a. Navigate to the Error Queue screen in the Mobile Manager.
   b. Click on the modified record.
   c. Click **Execute**.

> **Note:** For more information on the error queue and how to reapply the records using an API, see Section 3.8.3 Resolving Conflicts Using the Error Queue in the *Oracle Database Lite Developer's Guide*.

6.7 Monitoring and Analyzing Performance

The following sections describe how to monitor and analyze Data Synchronization performance.

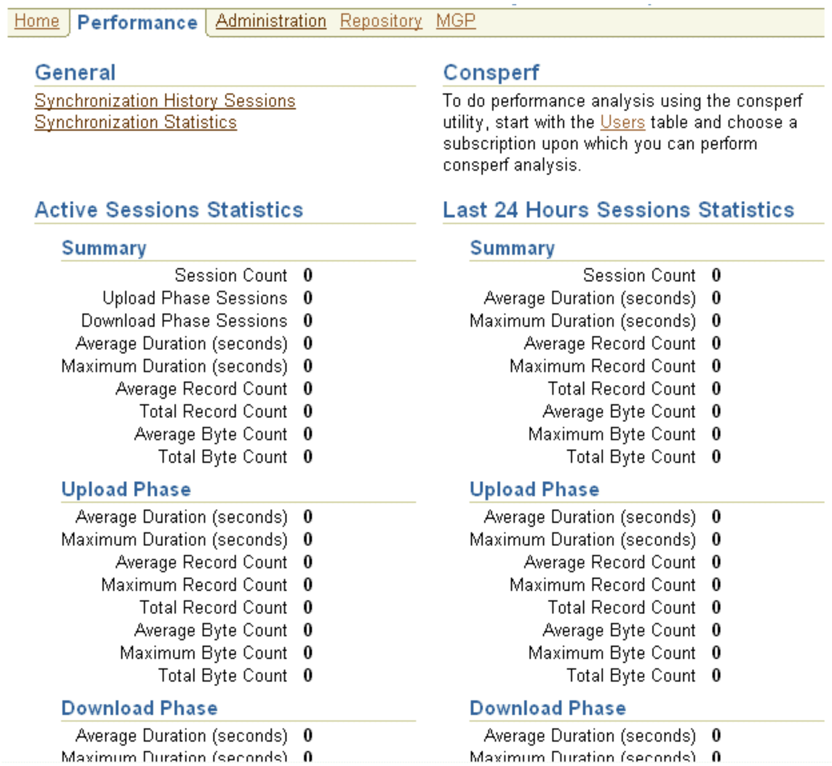
- Section 6.7.1, "Viewing Sync Server Statistics"
- Section 6.7.2, "Viewing MGP Cycles and Statistics"
- Section 6.7.3, "Analyzing Performance of Publications With the Consperf Utility"
- Section 6.7.4, "Monitoring Synchronization Using SQL Scripts"
- Section 6.7.5, "Synchronization Performance Affected by WebCache"

6.7.1 Viewing Sync Server Statistics

The Performance tab displays the Sync Server 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 6–8, you can see the active Sync Server statistics from the currently active sessions and compare it to overall statistics gathered from all sessions in the past 24 hours. This includes an overall section, the upload phase, and the download phase.

Figure 6–8 Performance Page



General

Synchronization History Sessions
Synchronization Statistics

Consperf

To do performance analysis using the consperf utility, start with the Users table and choose a subscription upon which you can perform consperf analysis.

Active Sessions Statistics

Summary

Session Count	0
Upload Phase Sessions	0
Download Phase Sessions	0
Average Duration (seconds)	0
Maximum Duration (seconds)	0
Average Record Count	0
Total Record Count	0
Average Byte Count	0
Total Byte Count	0

Upload Phase

Average Duration (seconds)	0
Maximum Duration (seconds)	0
Average Record Count	0
Maximum Record Count	0
Total Record Count	0
Average Byte Count	0
Maximum Byte Count	0
Total Byte Count	0

Download Phase

Average Duration (seconds)	0
Maximum Duration (seconds)	0

Last 24 Hours Sessions Statistics

Summary

Session Count	0
Average Duration (seconds)	0
Maximum Duration (seconds)	0
Average Record Count	0
Maximum Record Count	0
Total Record Count	0
Average Byte Count	0
Maximum Byte Count	0
Total Byte Count	0

Upload Phase

Average Duration (seconds)	0
Maximum Duration (seconds)	0
Average Record Count	0
Maximum Record Count	0
Total Record Count	0
Average Byte Count	0
Maximum Byte Count	0
Total Byte Count	0

Download Phase

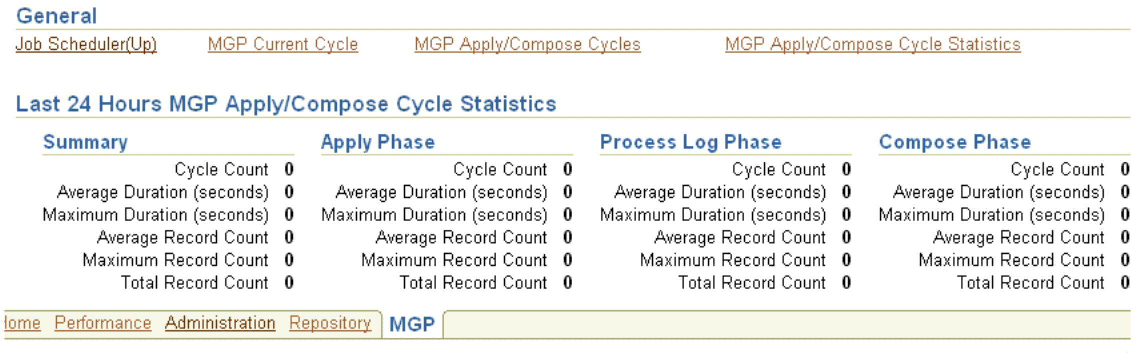
Average Duration (seconds)	0
Maximum Duration (seconds)	0

To view statistics from other dates, click the **Synchronization Statistics** link in the General section of this page. The Synchronization 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.

6.7.2 Viewing MGP Cycles and Statistics

By navigating to the MGP tab off of the Data Synchronization screen, the Mobile Server administrator can view the current status of the MGP cycle (see Figure 6–9). The columns are separated so that you can see how, in the last 24 hours, the MGP has performed overall, as well as for each individual phase: apply, compose and process.

Figure 6–9 MGP Page



Last 24 Hours MGP Apply/Compose Cycle Statistics

Summary		Apply Phase		Process Log Phase		Compose Phase	
Cycle Count	0	Cycle Count	0	Cycle Count	0	Cycle Count	0
Average Duration (seconds)	0	Average Duration (seconds)	0	Average Duration (seconds)	0	Average Duration (seconds)	0
Maximum Duration (seconds)	0	Maximum Duration (seconds)	0	Maximum Duration (seconds)	0	Maximum Duration (seconds)	0
Average Record Count	0	Average Record Count	0	Average Record Count	0	Average Record Count	0
Maximum Record Count	0	Maximum Record Count	0	Maximum Record Count	0	Maximum Record Count	0
Total Record Count	0	Total Record Count	0	Total Record Count	0	Total Record Count	0

When you click on MGP Current Cycle, you can see what the MGP process is currently doing. For instance, you can check if the apply or compose cycle is running when the MGP cycle is in progress. If you have set the `MGP_HISTORY` instance parameter, (see Section 6.3, "Configuring Data Synchronization"), then upon completion of the apply or compose cycle, the cycle details are stored in Cycle History.

Since the front page only shows the last 24 hours, you can view farther back by clicking on the MGP Apply/Compose Cycle Statistics. You can set a date range to search and can even specify whether to search based upon the following:

- Apply Only or Apply/Compose
- Success, Failure, or Conflict results

When you click MGP Apply/Compose cycles, you can search for a range of historical records of these cycles and then view the details of each cycle.

6.7.3 Analyzing Performance of Publications With the Consperf Utility

The Consperf utility profiles your subscriptions and may modify how the publication item is executed if the utility determines that there is a more performant option. The Consperf tool evaluates how the SQL within the publication item interacts with our Data Synchronization query templates. The first synchronization is always a complete refresh, which is a direct invocation of the query. On subsequent synchronizations, the query templates determine incremental refreshes. This improves your performance from not having to perform a complete refresh each time you synchronize. However, the interaction of our query templates and your SQL may not be optimal, which is discovered by the Consperf tool. We either modify the query template or type of logical delete or insert for you or you can adjust your SQL to be more performant in regards to our templates.

In addition, application developers and administrators use this utility to analyze the performance of subscriptions and identify potential bottlenecks during synchronization.

This tool generates the following two primary analysis reports:

1. Timing statistics for publication items
2. Explain plans for publications

The Consperf tool automatically tunes subscription properties, if the default templates do not supply the highest performing option. You can select a client and choose the desired subscription for performance analysis. Users can change parameter values before analyzing performance. The analysis results, which are timing and execution plan reports, are stored on the server and can be accessed by viewing the same user and subscription.

You can execute the Consperf utility through one of the following locations:

- Click the Users link under the Consperf section on the Performance tab.
- Click the Users link from the Repository screen.

Then, perform the following:

1. Select the User that you want to execute the Consperf tool against and click **Subscriptions**.
2. From the subscriptions screen, choose the publication and click **Consperf performance analysis**. This starts the Consperf analysis.
3. Click **Set consperf parameters and launch the consperf thread**, which brings you to a screen where you can configure parameters that effect how the performance analysis is executed. See Section 6.7.3.1, "Deciphering the Performance Evaluation Files" for more information on these parameters and how they effect the performance evaluation output.
4. Once you have set the configuration for how you want your performance analysis to occur, click **OK**. The Consperf tool executes and prepares the reports for you, based on your configuration. You are returned to the first Consperf page with the reports listed as hyperlinks under the **Last Consperf Run Results** section as **View Timing File** or **View Execution Plan File**.

6.7.3.1 Deciphering the Performance Evaluation Files

There are two performance evaluatons that come out of the Consperf utility:

- Timing File
- Execution Plan File

Timing File

The timing file contains the analysis of how the publication item performs with the data synchronization defaults against how it could perform if other options were chosen. The output of this file shows you the conclusions of the analysis and how the data synchronization defaults could be modified to perform better with your particular publication items.

The first section of the timing file provides you information on the configuration with which this analysis was executed. Thus, if you modify the configuration for other analysis, you can go back and compare each file to each other to easily see the differences in the output.

> **Note:** The results of this analysis may cause the data synchronization engine to modify the type of query template or logical delete/insert/update used with your publication item. To change it back to the defaults, you will have to rerun Consperf with `CLEARTUNE` set to `YES`. See Table 6–3 for a full description of parameter settings.

The following example shows the publication that is examined is the `T_SAMPLE11` publication. The version of the Oracle Database Lite is 10.0.0.0.0. The user is `S11U1`. And the configuration is set to time out if the query takes longer that 1000 milliseconds and change the defaults if the difference between the default and the other templates are greater than 20 seconds (20000 milliseconds). The command that authorizes the changes is when AUTOTUNE is set to true. If set to false, the analysis is provided, but nothing is modified.

```
VERSION = 10.0.0.0.0
OPTMIZER_MODE = null
APPLICATION = null
PUBLICATION = T_SAMPLE11
CLIENTID = S11U1
TIMEOUT = 1000 ms
TOLERANCE = 20000 ms
ITERATIONS = 2
AUTOTUNE_SUPPORT = true
```

The next part of the Timing File lists the time in milliseconds each template type takes to complete with each publication item in the publication. There are three templates that data synchronization can use to "wrap" your SQL query. The default query template is SYNC_1. Since the tolerance is set to 20 seconds, then if either template SYNC_2 or SYNC_3 perform at least 20 seconds bettern than SYNC_1, then the template type will be modified for your publication item. You can set the TOLERANCE level to fewer seconds in the Consperf configuration. See Table 6–3 for a description of TOLERANCE.

```
Publication Item Name | NS | BS | SYNC_1 | SYNC_2 | SYNC_3 | AS | Total
 -----------------------------------------------------------------------
P_SAMPLE11-D          | <3> | <0> | <6>  | 10     | -1000  | <0> | 9
P_SAMPLE11-M          | <3> | <0> | <5>  | 8      | -1000  | <0> | 8
```

- There are two publication items in the subscription.
- NS stands for Null Sync. Your application may be issuing a null synchronization. If so, this shows the time in milliseconds that it took to complete. The null synchronization is a tool to see if it is the data that is causing the performance hit or the application itself.
- BS stands for Before Synchronization; AS stands for After Synchronization. You can provide callouts that are executed either before or after each synchronization for this application. This shows the time in milliseconds it takes to perform each task. In this example, there is no before or after synchronization callouts.
- SYNC_1 is the default template. In combination with the publication items, it still is executing the fastest as compared to the other two options: SYNC_2 and SYNC_3 with 6 and 5 milliseconds for each publication item respectively. Thus, these publication items will continue to use SYNC_1 template. Note that SYNC_3 has -1000 as its time. That either means that the template was not appropriate to execute or that it timed out.
  - SYNC_1 uses an outer-join for inserts, updates, and deletes
  - SYNC_2 is a simple insert and update
  - SYNC_3 uses the base view for insert and update. The base view is the first table in the select statement, as it is the primary key used to search for all records in the query.
- The total is the total number of milliseconds to execute the entire publication item.

The second section is how the MGP performs with the templates it uses for deletes and inserts. It evaluates the default against other options, as follows:

- Logical delete options:

  - MGP template for logical deletes using EXISTS: default for logical delete
  - MGP template for logical deletes using correlated IN
  - MGP template for logical deletes using HASH_AJ
  - MGP template for logical deletes using IN
- Logical insert options:
  - MGP template for logical inserts using EXISTS: default for logical insert
  - MGP template for logical inserts using correlated IN
  - MGP template for logical inserts using IN
- Logical update options
  - MGP template for logical updates using correlated IN: default for logical updates
  - MGP template for logical updates using EXISTS
  - MGP template for logical updates using IN
- MGP template for logical updates with multiple table dependencies

For example, the following evaluates how each publication item performs with its logical deletes:

```
MGP Output...
Pub Item Name | LDEL_1 | LDEL_2 | LDEL_3 | LDEL_4
P_SAMPLE11-D  | <5>    | 3      | 3      | 3
P_SAMPLE11-M  | <5>    | 3      | 5      | 4
```

The LDEL_1 is the default and even though LDEL_2 , 3 and 4 are faster, they are not 20 seconds faster, which is the tolerance level. So, the default for deletes is kept the same. If the difference in speed had been greater than the tolerance level, the Consperf utility would have modified the logical delete method in the repository for the publication item in future—if the autotune parameter was set to yes.

The last section, Subscription Properties, describes the following:

- Profiled: Has autotune been turned on and Consperf executed previously on this subscription?
- Base View: True if this publication item uses more than one table.
- How many records are in the subscription.
- How many records are dirty?
- How many records have been flagged as dirty to simulate an actual run? Up to the number of records in the subscription or MAXLOG will be flagged as dirty, whichever is least.

Configuration for Data Synchronization

Table 6–2 Consperf Parameters for Both Synchronization and MGP Processing

Parameter	Default Value	Allowed Values	Description
PUBITEMLIST	<ALL>	Pub1, Pub2, and so on.	Specifies comma-separated list of publication items to process. The default is all publication items in the publication.

Table 6–2 (Cont.) Consperf Parameters for Both Synchronization and MGP Processing

Parameter	Default Value	Allowed Values	Description
SKIPPUBITEMLIST	<NONE>	Pub1, Pub2, and so on.	Specifies comma-separated list of publication items to skip.
OPTIMIZER	<DB>	Can set to RULE or CHOOSE; otherwise sets to what database is set to.	Specifies the optimizer mode to use within Oracle. The default is the current DB setting.
ORDERBYPUBITEM	NO	Yes or No	Orders all output by publication item name.

Table 6–3 Consperf Parameters for Synchronization Timing Performance

Parameter	Default Value	Allowed Values	Description
TIMEOUT	10 seconds	Integer for seconds	Specifies the query timeout value in seconds. This is the amount of time Consperf will wait before it cancels a query.
UPDATECOUNT	5	Integer for number of records	Specifies the number of records to mark as dirty during synchronization.
MAXLOG	5000	Integer for number of records	Specifies the number of records to put in the log table. Simulates the transaction log
AUTOTUNE	NO	Yes or No	Enables auto-tune.
CLEARTUNE	NO	Yes or No	Clears existing auto-tune results.
TOLERANCE	20 seconds	Integer for seconds	A template must be faster by this number of seconds before it replaces the default template.

Execution Plan File

The exeuction plan file shows how your publication items interact with the different logical delete, insert, and update templates. From this report, you can evaluate your SQL to see if you want to modify it in any way to speed up your query. Set the optimizer parameter to designate how the database is organized. If you set this parameter to a setting that the database is not set to, it still acts as if the database is set to this way to show you how it would execute. See Table 6–4 for all configuration parameters that relate to this search.

Table 6–4 Consperf Parameters for Execution Performance Plan

Parameter	Default Value	Allowed Values	Description
GATHERSTATS	NO	Yes or No	Gathers optimizer statistics on all mobile server objects. MGP compose MUST be disabled while Consperf analyzes objects. Consperf blocks this automatically, but the safest approach is to manually stop the MGP process before running Consperf with the GATHERSTATS option. If Consperf fails while gathering statistics, users must re-run CLEARSTATS before starting the MGP process again.
CLEARSTATS	NO	Yes or No	Removes optimizer statistics on mobile server objects.

Table 6–4 (Cont.) Consperf Parameters for Execution Performance Plan

Parameter	Default Value	Allowed Values	Description
SQLTRACE	NO	Yes or No	Enables Oracle sql trace. TKPROF can be used to analyze the resulting trace file.

6.7.4 Monitoring Synchronization Using SQL Scripts

If, instead of viewing MGP statistics within the Mobile Manager, you would rather execute SQL scripts to monitor Mobile application status during synchronization, you may use any of the following SQL scripts to retrieve the desired information.

- Section 6.7.4.1, "Synchronization Times for All Clients"
- Section 6.7.4.2, "Failed Transactions for all Clients"
- Section 6.7.4.3, "Completely Refreshed Publication Items for all Clients"
- Section 6.7.4.4, "Publications Flagged for Complete Refresh for All Clients"
- Section 6.7.4.5, "Clients and Publication where Subscription Parameters are Not Set"
- Section 6.7.4.6, "Record Counts for Map-based Publication Item by Client"
- Section 6.7.4.7, "Record Count for Map-based Publication Items by Store"
- Section 6.7.4.8, "All Client Sequence Partitions and Sequence Values"
- Section 6.7.4.9, "All Publication Item Indexes"

6.7.4.1 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
/
```

6.7.4.2 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
/
```

6.7.4.3 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
/
```

6.7.4.4 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'
/
```

6.7.4.5 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
/
```

6.7.4.6 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
/
```

6.7.4.7 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
/
```

6.7.4.8 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
/
```

6.7.4.9 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
/
```

6.7.5 Synchronization Performance Affected by WebCache

The amount of time it takes to complete a synchronization for a Web-to-Go client is significantly increased when WebCache is installed. See Section 13.5, "Synchronization Performance Affected by WebCache" for more information.

7

Job Scheduler

The following sections describe how you can manage the job engine and schedule jobs through the Mobile Manager.

- Section 7.1, "Scheduling a Job to Execute at a Specific Time or Interval"
- Section 7.2, "Managing the Job Engine"
- Section 7.3, "Manage Scheduled Jobs Using the Mobile Manager"
- Section 7.4, "Managing Scheduled Jobs Using ConsolidatorManager APIs"
- Section 7.5, "Using the ConsolidatorManager APIs to Create Jobs"

7.1 Scheduling a Job to Execute at a Specific Time or Interval

You can choose to execute any job—which can be any application—at a specific time or interval. For example, the default MGP process (see Section 6.1, "How Does the Synchronization Process Work?") is a job that executes at a regular interval. The default behavior is for the MGP process to execute every 60 seconds to apply all incoming modifications from the clients and compose all outgoing messages to the clients from the repository. You can define how often the MGP process executes, or even schedule a time for it to stop execution. You can schedule any job with this same mindset.

Note: For an overview on how to create a job out of one of your applications, see Section 7.5, "Using the ConsolidatorManager APIs to Create Jobs".

The engine that monitors the job execution is the Job Scheduler. For example, by default, the Job Scheduler fires off the MGP process every 60 seconds. It is the mechanism that tracks all of the scheduled jobs and ensures that your defined job is executed when you wanted it to be executed. You can turn it on and off, and monitor alerts specific to the Job Scheduler.

See Section 7.2, "Managing the Job Engine" for details on how to manage the job engine; see Section 7.3, "Manage Scheduled Jobs Using the Mobile Manager" on how to create and manage jobs that you want scheduled.

Note: Within the Web-to-Go client, you can also schedule when a synchronization is started on the client. This is separate from the Mobile Server Job Scheduler. See Section 3.1.1.2, "Configuration Tab" for more information.

7.2 Managing the Job Engine

The Job Scheduler manages jobs that you create and schedule. However, the Job Scheduler needs to be managed, as well. You navigate to the Job Scheduler page in the Mobile Manager by selecting a Mobile Server from the list of Mobile Servers on the Mobile farm screen. At the bottom of the Mobile Server screen, click **Job Scheduler**, which brings up the Home screen for the Job Scheduler.

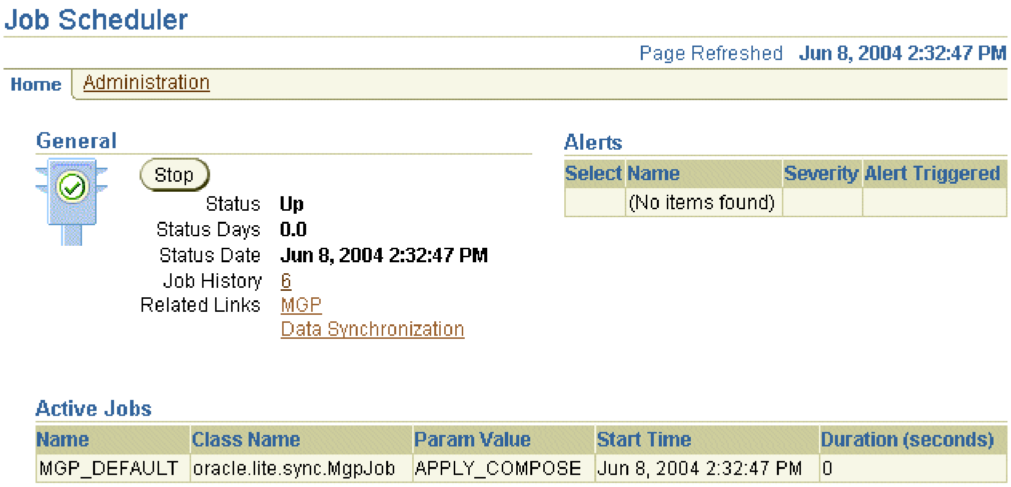
The following sections describe how to manage the Job Scheduler:

- Section 7.2.1, "Starting the Job Scheduler"
- Section 7.2.2, "Checking Job Scheduler Alerts"
- Section 7.2.3, "Managing Active Jobs"
- Section 7.2.4, "Managing the Job History List"

7.2.1 Starting the Job Scheduler

Figure 7–1 displays the Job Scheduler's default status on the Job Scheduler home page. 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 7–1 The Job Scheduler Home Page



Page Refreshed Jun 8, 2004 2:32:47 PM

General

Status	Up
Status Days	0.0
Status Date	Jun 8, 2004 2:32:47 PM
Job History	6
Related Links	MGP Data Synchronization

Alerts

Select	Name	Severity	Alert Triggered
	(No items found)		

Active Jobs

Name	Class Name	Param Value	Start Time	Duration (seconds)
MGP_DEFAULT	oracle.lite.sync.MgpJob	APPLY_COMPOSE	Jun 8, 2004 2:32:47 PM	0

To stop the Job Scheduler, click **Stop**. Stopping the Job Scheduler prevents any new jobs from starting. However, any existing jobs will continue to execute until finished. Stopping the Job Scheduler does not kill any existing jobs. If you want to prevent only a single job from being launched, then disable the application on the Administration screen. See Section 7.3.3, "Enabling or Disabling Jobs" for more information on disabling applications.

7.2.2 Checking Job Scheduler Alerts

When the Job engine fails, then the Alerts table displays these exceptions as critical alerts. When the Job engine has trouble with executing your job, then these exceptions are displayed as warning 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**.

7.2.3 Managing Active Jobs

As shown in Figure 7–1, 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.

For more information on how to manage jobs, see Section 7.3, "Manage Scheduled Jobs Using the Mobile Manager".

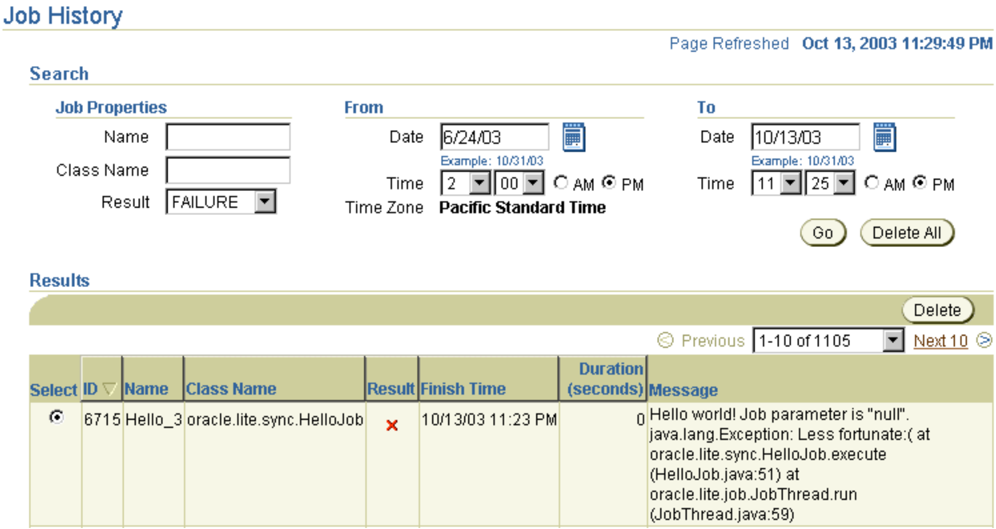
To terminate a job, click the **Administration** tab, select the job, and click **Delete**. This does not terminate active jobs, but prevents the job from executing in the future.

7.2.4 Managing the Job History List

The number of current registered jobs in the Job History list is listed on the Job Scheduler home page under the Status Date line. Click on the number displayed to bring up the Job History page. The Job History page enables you to provide criteria to search, sort, and manage the job history based on job properties—such as name, class name, result, or a specific date and time. Based on your search criteria, the Job History page displays job history details under the **Results** section.

Figure 7–2 displays the Job History page.

Figure 7–2 Job History Page



Job History

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

Search

Job Properties

Name

Class Name

Result FAILURE

From

Date 6/24/03

Example: 10/31/03

Time 2 00 ○ AM ● PM

Time Zone Pacific Standard Time

To

Date 10/13/03

Example: 10/31/03

Time 11 25 ○ AM ● PM

Results

1-10 of 1105

Select	ID	Name	Class Name	Result	Finish Time	Duration (seconds)	Message
●	6715	Hello_3	oracle.lite.sync.HelloJob	✗	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)

You can sort the messages by any of the headers. For example, to sort the job history details by name, click **Name** in the header title region. It toggles between A-Z and Z-A.

To delete a single job, select the job and click **Delete**. To delete all job history entries that match your search criteria, click **Search and Delete**.

7.3 Manage Scheduled Jobs Using the Mobile Manager

The most notable scheduled job is the MGP (see Section 6.1, "How Does the Synchronization Process Work?"). By default, it is scheduled to execute every 60 seconds to perform a specific task for data synchronization. You can modify the schedule of this existing job, as well as create other jobs for your own purposes to execute at a regular interval.

From the Job Scheduler screen, click the Administration tab, where you can create a new job or edit existing jobs. The Scheduled Jobs section displays the jobs that are scheduled in the job engine.

The following sections enable administrators to accomplish the following tasks:

- Section 7.3.1, "Creating a New Job"
- Section 7.3.2, "Editing Existing Jobs"
- Section 7.3.3, "Enabling or Disabling Jobs"
- Section 7.3.4, "Deleting Jobs"
- Section 7.3.5, "Default Jobs"

7.3.1 Creating a New Job

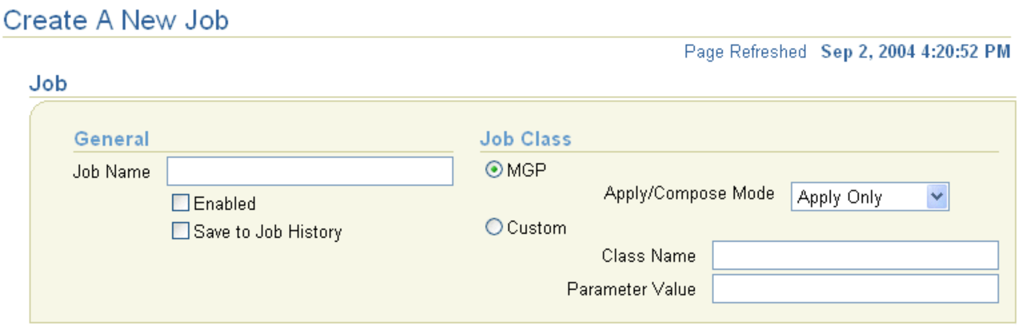
In order to create a new job, you create the schedule of how often and when an existing application is executed. To create this schedule, navigate to the Job Scheduler Administration screen and click **Create A New Job**.

Figure 7–3 displays the top section of the Create a New Job page. Give the job a name, select the checkbox for Enabled to enable the job (or leave blank to leave disabled), and select the checkbox for Save to Job History if you want a record of this application executing.

Under the Job Class section, if this is for the MGP process, then select if this is for Apply Only or Apply and Compose. The MGP process can be modified to perform only application of new and modified records from the clients. This is beneficial for applications that never have to update information from the back-end server database. Choosing Apply Only saves performance if it is relevant for your application. For example, if you had a company that performed a lot of updates throughout the day, but no one needed to know the new information until the next day, you could schedule an MGP process to perform Apply Only all day to update the repository, and schedule another MGP process that executes only at night with Apply/Compose to perform the last updates and then bring down all of the days modifications to all of the users.

If this is not for an MGP process, then enter the class name to be executed for this job and any parameter values. Since you design the class, enter the parameter as you have designed the parameter format. There are two default jobs, which are described in Section 7.3.5, "Default Jobs".

Figure 7–3 Create a New Job - Top Section



Create A New Job

Page Refreshed **Sep 2, 2004 4:20:52 PM**

Job

General

Job Name

□ Enabled

□ Save to Job History

Job Class

● MGP

Apply/Compose Mode: Apply Only

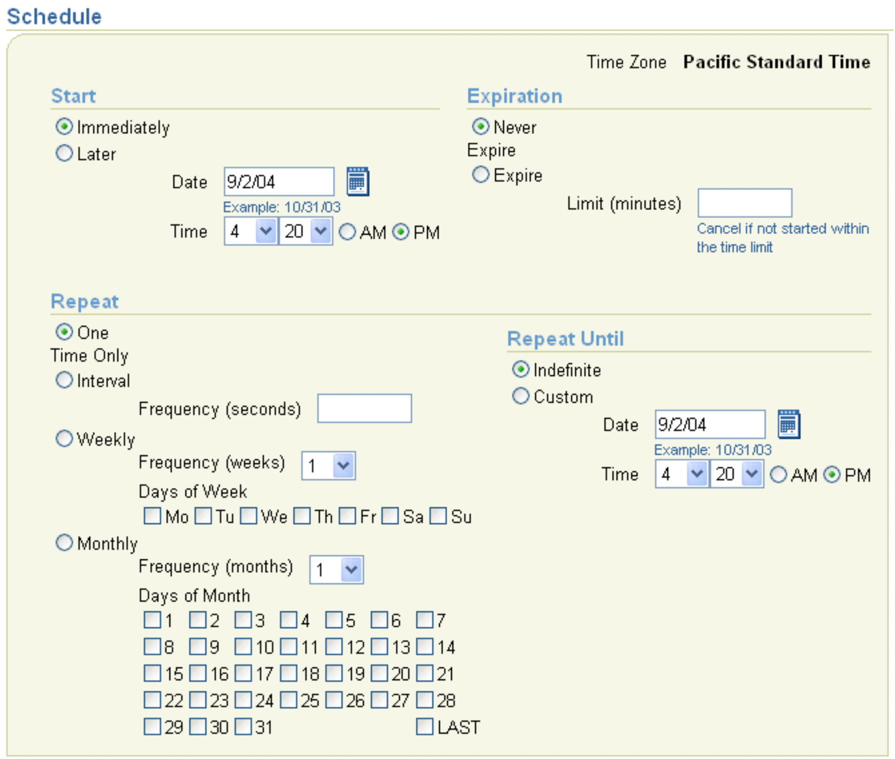
○ Custom

Class Name

Parameter Value

Figure 7–4 displays the bottom section of the Create a New Job page, which is where you define when and how often your job executes.

Figure 7–4 Create a New Job - Bottom Section



Schedule

Time Zone **Pacific Standard Time**

Start

● Immediately
○ Later

Date 9/2/04
Example: 10/31/03
Time 4 20 ○ AM ● PM

Expiration

● Never Expire
○ Expire

Limit (minutes)
Cancel if not started within the time limit

Repeat

● One Time Only
○ Interval

Frequency (seconds)

○ Weekly

Frequency (weeks) 1

Days of Week

□ Mo □ Tu □ We □ Th □ Fr □ Sa □ Su

○ Monthly

Frequency (months) 1

Days of Month

□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
□ 8 □ 9 □ 10 □ 11 □ 12 □ 13 □ 14
□ 15 □ 16 □ 17 □ 18 □ 19 □ 20 □ 21
□ 22 □ 23 □ 24 □ 25 □ 26 □ 27 □ 28
□ 29 □ 30 □ 31 □ LAST

Repeat Until

● Indefinite
○ Custom

Date 9/2/04
Example: 10/31/03
Time 4 20 ○ AM ● PM

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

Table 7–1 describes data that must be entered in the **Start** section.

Table 7–1 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 and specify the date and time when this job is to start.	Optional

Table 7–2 describes data that must be entered in the **Expiration** section.

Table 7–2 Expiration Details Description - Schedule Section

Field	Description	Required
Never Expire	To ensure that the chosen job schedule does not expire—that is, this job always executes—select this option.	Optional
Expire	If you want the job to expire after a specified number of minutes—even if it has not execute yet—then specify the number of minutes in this field. The Job Scheduler cancels jobs that do not start at the specified time. However, it does not stop jobs that have already started.	Optional

Table 7–3 describes how often the job executes in the **Repeat** section.

Table 7–3 Repeat Details Description - Repeat Section

Field	Description	Required
One Time Only	The job executes only once.	Optional
Interval	The job executes after a specified interval has passed. The interval duration between execution of the job is defined in seconds.	Optional
Weekly	The job executes on the specified day of the week. You can specify an interval of whether this executes weekly (1 in the Frequency pulldown), every other week (2 in the Frequency pulldown) and so on.	Optional
Monthly	The job executes on a specified day of the month. Same as above, but with the months of the year.	Optional

Table 7–4 defines whether the chosen schedule repeats indefinitely or whether you want it to execute only on a certain date/time.

Table 7–4 Repeat Until Details Description - Repeat Section

Field	Description	Required
Indefinite	To repeat the job schedule indefinitely, select this option.	Optional
Custom	To specify how long this job executes until, specify the date and time of when the job is canceled.	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.

7.3.2 Editing Existing Jobs

Navigate to the Job Scheduler Administration screen. To edit existing jobs, click **Edit**. Modify the same fields that are described in Section 7.3.1, "Creating a New Job".

7.3.3 Enabling or Disabling Jobs

You can enable or disable a job from the Administration screen off of the main Job Scheduler screen. Select the job that you need to modify and either click **Enable** or **Disable**. The **Status** column confirms the changed status.

7.3.4 Deleting Jobs

Navigate to the Job Scheduler Administration screen. 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.

7.3.5 Default Jobs

The Oracle Database Lite 10*g* 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 Process: `MGP_DEFAULT`
- Purging History: `PURGE_HISTORY_DEFAULT`

7.3.5.1 MGP_DEFAULT

You have to have at least a single MGP process for apply/compose phase of the synchronization phase. The MGP_DEFAULT is this process. You can modify this process to be apply only, or you can modify when the MGP process is executed. You can create other MGP processes, if you wish.

job Name

`MGP_DEFAULT`

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 list.

7.3.5.2 PURGE_HISTORY_DEFAULT

In order to preserve disk space, the administrator wants to purge the history. This job is created for you to automatically purge the history at a selected interval. You can modify the interval or disable this job, if you wish. This section describes the job class, job parameter value and its corresponding description.

Job Name

`PURGE_HISTORY_DEFAULT`

Job Class

`oracle.lite.sync.PurgeHistoryJob`

Job Parameter Value

`History=Sync,MGP,Job;Days=7`

Since this Job is a customized class, the parameter is defined and parsed within the purge history class. The structure of this parameter is a string with two name/value pairs: what type of history to purge and for how many days. In this example, the history purged is for the Sync, MGP, and Job historical data. The history is purged for the last seven days. You can modify the number of days or add/delete the history logs that this applies to. The only options are Sync, MGP, or Job. For example, if you want every record that is 3 days old or more to be erased, modify the 7 to a 3.

7.4 Managing Scheduled Jobs Using ConsolidatorManager APIs

Application developers can define, submit, and manage jobs programmatically. The Job Scheduler provides an API 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 API comprises some methods of the `oracle.lite.sync.ConsolidatorManager` class 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 to the *Oracle Database Lite API JavaDoc.*

7.5 Using the ConsolidatorManager APIs to Create Jobs

Within the `oracle.lite.sync.ConsolidatorManager` class, there are several APIs, which are documented fully in the *Oracle Database Lite API JavaDoc*, that enable you to create, register, and schedule your job.

While these methods are described fully in the JavaDoc, the following demonstrates the order in which you would execute the methods:

1. Create your job class by implementing the `oracle.lite.job.Job` interface. Implement the Job interface methods, as follows:
   - `init` method—This method is invoked by the Job Scheduler when the job is loaded.
   - `execute` method—This method is invoked by the Job Scheduler when the job is scheduled to execute. Put a call into your application within this method. The Job Scheduler passes in the input parameter that was provided when the job is created—either with the `createJob` method or within the Mobile Manager Job Scheduler screen. When finished, the `execute` method returns an object of class type `ExecutionResult` containing whether the job was a success or failure.
   - `destroy` method—This method is invoked after the job completes.
2. After you have created your job class, register it with the `registerJobClass` method.
3. Create the job in the Job Scheduler by executing the `createJob` method. One of the input parameters is an object of class type `Schedule`, which defines when the job is executed. There are also other management methods that correspond to the Mobile Manager GUI, such as `dropJob`, `enableJob`, and `disableJob`.
4. If you want to retrieve any logs, execute the `getJobExecutionLogs` method, which retrieves objects of `ExecutionLog` class.

8 Manage Your Devices

When you install your Mobile client software, the Mobile device management client software is automatically installed and, in most cases, bootstrapped. Within the Mobile Manager, the administrator can send commands to remote devices. The next time that the device is available—either through wireless connection or synchronization—the command that you send will execute.

The following sections describe how to manage your devices:

- Section 8.1, "Installing the Mobile Client Software on Your Mobile Device"
- Section 8.2, "Modifying Environment on Your Mobile Clients During Installation"
- Section 8.3, "Viewing Device Information"
- Section 8.4, "Configuring and Customizing Your Mobile Device Platform"
- Section 8.5, "Configuring Your Mobile Devices"
- Section 8.6, "Sending Commands to Your Mobile Devices"
- Section 8.7, "Enabling Device Software Updates"
- Section 8.8, "Using the Device Management Client GUI to Manage Device on Client-side"
- Section 8.9, "Managing the Network Protocol Between the Device and the Mobile Client Software"
- Section 8.10, "Installation Configuration (INF) File"
- Section 8.11, "Defining Device Manager Commands With the Device Manager OTL Tag Language"

8.1 Installing the Mobile Client Software on Your Mobile Device

The Mobile device software for your language and platform is installed when you install the Mobile client on your device. See Chapter 2, "Installing Mobile Clients" for more information.

You can customize the installation for your Mobile clients by customizing the platform and the setup configuration files for the platform, as follows:

- Certain modifications can be made to the Mobile client configuration files during installation. See Section 8.2, "Modifying Environment on Your Mobile Clients During Installation" for more information.
- Customize the platform to install additional binaries, applications, and other environment modifications. See Section 8.4, "Configuring and Customizing Your Mobile Device Platform" for more information.

Once installed, you can manage the client using the Oracle Lite Device Manager tool. See Section 8.8, "Using the Device Management Client GUI to Manage Device on Client-side" for more information.

8.2 Modifying Environment on Your Mobile Clients During Installation

The Mobile client installs a few configuration files on your clients. These configuration files are discussed more in the *Oracle Database Lite Developer's Guide*.However, you can add name/value pairs to the `webtogo.ora`, `POLITE.INI` and `ODBC.INI` files on the client file, which are available in Windows under `%WINDIR%` directory and in Linux under `$OLITE_HOME/bin` directory, by modifying the INF file corresponding to the Oracle Database Lite platform that is to be downloaded to the client. You can add the new configuration directly in the `<ini>` section of the INF file by editing the INF file yourself or you can add them through the Mobile Manager. To edit the INF file directly, see Section 8.10, "Installation Configuration (INF) File".

To edit and add name/value pairs for the `webtogo.ora`, `POLITE.INI`, and/or `ODBC.INI` files, do the following:

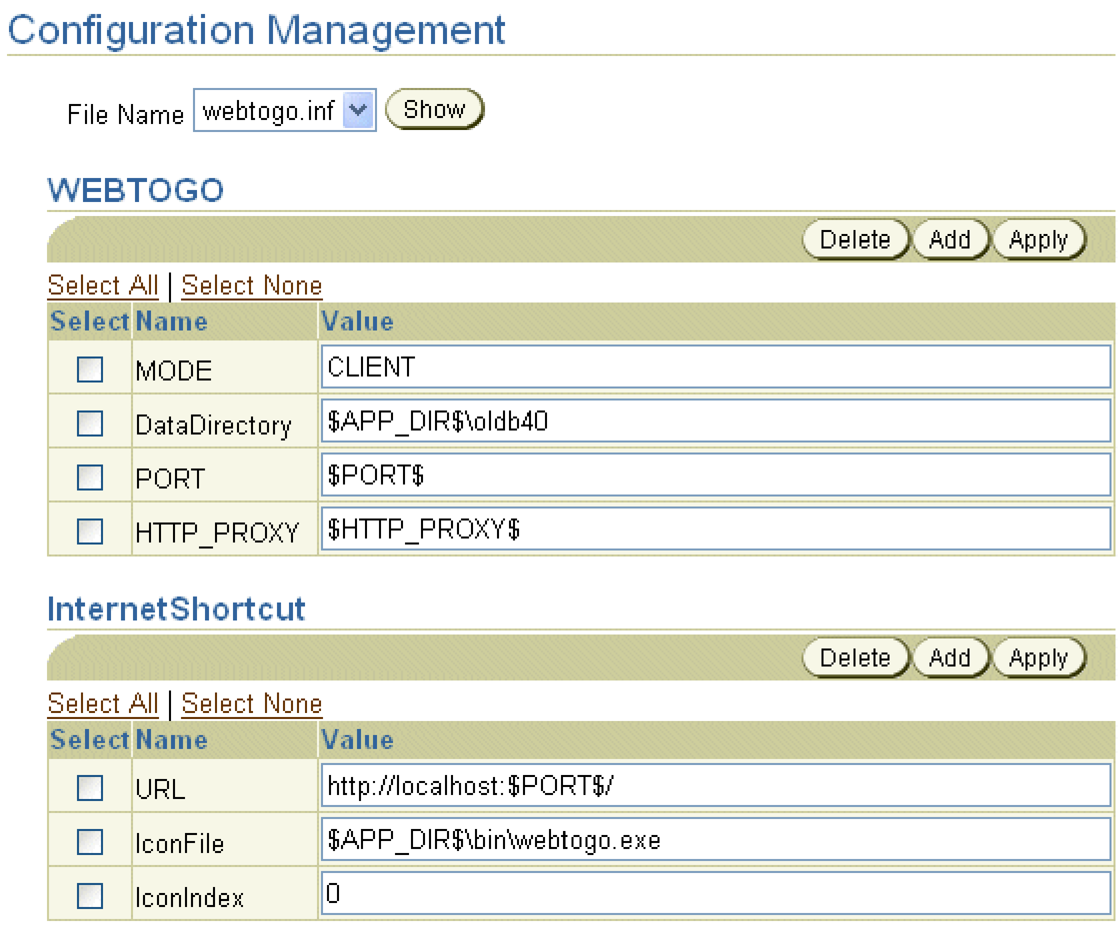
1. Navigate to the Mobile Device screen.
2. Click **Administration**.
3. Click **Configuration Management**.
4. Select the INF file from the File Name pull down and click **Show**. This enables you to modify the INI section in this particular INF file. All of the current assignments are displayed.

   For each INF file, the INI section is displayed. You can only add name value pairs to existing sections. For example, the following is the INI section from the `webtogo.inf` file:

   ```
   <ini>
     <item name="$APP_DIR$\bin\webtogo.ora" section="WEBTOGO">
       <item name="MODE">CLIENT</item>
       <item name="DataDirectory">$APP_DIR$\oldb40</item>
       <item name="PORT" replace="false">$PORT$</item>
       <item name="HTTP_PROXY" replace="false">$HTTP_PROXY$</item>
     </item>
     <item name="$DESKTOP$\Web-to-Go.url" section="InternetShortcut">
        <item name="URL" replace="false">http://localhost:$PORT$/</item>
        <item name="IconFile">$APP_DIR$\bin\webtogo.exe</item>
        <item name="IconIndex">0</item>
     </item>
   </ini>
   ```

   There are two sections configured in this INI section: `WEBTOGO` and `InternetShortcut`. You can add other sections. The Mobile Manager notes if there are sub-items and will create a heading for the sections. To add more sections, you must modify the INF file directly. To modify or add items to the existing sections, you can click **Add**. Figure 8–1 displays how the previous INI section is displayed in the Mobile Manager. To add a name/value pair, click Add. To modify a parameter, modify it and click Apply. To delete, select the configuration pair and click Delete.

Figure 8–1 Adding Name/Value Pairs to INF Files



Configuration Management

File Name: webtogo.inf

WEBTOGO

Select All | Select None

Select	Name	Value
□	MODE	CLIENT
□	DataDirectory	APP_DIR\oldb40
□	PORT	$PORT$
□	HTTP_PROXY	$HTTP_PROXY$

InternetShortcut

Select All | Select None

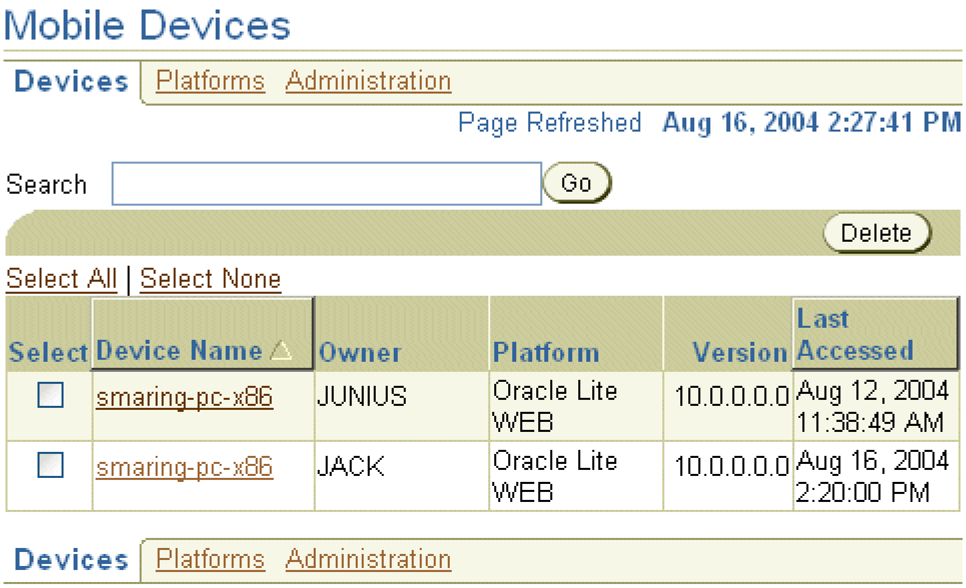
Select	Name	Value
□	URL	http://localhost:$PORT$/
□	IconFile	APP_DIR\bin\webtogo.exe
□	IconIndex	0

5. To add name/value pairs to the existing sections, click **Add**.
6. A screen is displayed asking for two strings: a name and a value. Enter these items and click **OK**.

8.3 Viewing Device Information

From the Mobile Devices screen, as shown inFigure 8–2, select the device that you want to know more about.

Figure 8–2 Devices Page



Mobile Devices

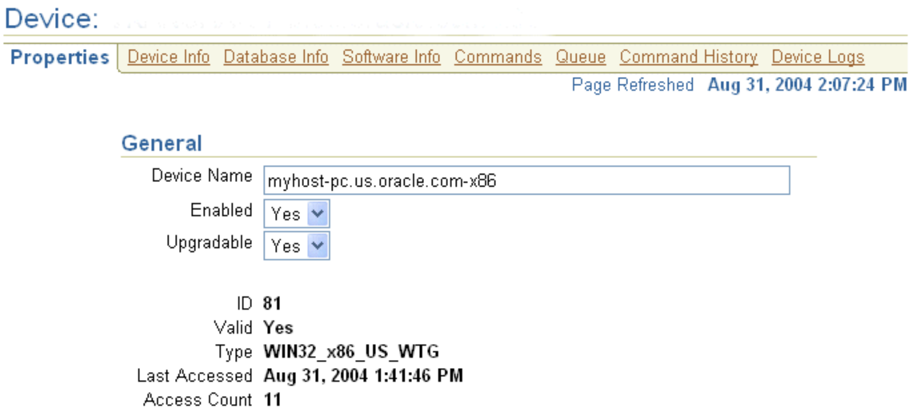
Page Refreshed **Aug 16, 2004 2:27:41 PM**

Search

Select	Device Name △	Owner	Platform	Version	Last Accessed
□	smaring-pc-x86	JUNIUS	Oracle Lite WEB	10.0.0.0.0	Aug 12, 2004 11:38:49 AM
□	smaring-pc-x86	JACK	Oracle Lite WEB	10.0.0.0.0	Aug 16, 2004 2:20:00 PM

Figure 8–3 shows the specific device page that comes up when you select the device that you are interested in. The Properties screen is the first set of information available.

Figure 8–3 Mobile Device Properties and Information



Device:

Page Refreshed **Aug 31, 2004 2:07:24 PM**

General

Device Name	myhost-pc.us.oracle.com-x86
Enabled	Yes
Upgradable	Yes
ID	**81**
Valid	**Yes**
Type	**WIN32_x86_US_WTG**
Last Accessed	**Aug 31, 2004 1:41:46 PM**
Access Count	**11**

On the Mobile Device Properties screen, you are told the following about the device:

- Device Name
- Enabled: If the device is enabled. See Section 8.4.2, "Enabling or Disabling All Mobile Devices in a Platform" for more information.
- Upgradable: If the device accepts software upgrades. See Section 8.4.3, "Allowing Software Upgrades to All Mobile Devices in a Platform" for more information.
- Valid: If the Device Management software is installed and correct. If de-installed, Valid displays No.

Each of the tabs at the top provides more information about the device. Use this information to determine what sort of administration each one needs in order to continue to operate smoothly. You can use this information to tell what needs to be upgraded on each device. The following sections covers each of these tabs.

- Section 8.3.1, "Viewing Device Information"
- Section 8.3.2, "Viewing Database Information"
- Section 8.3.3, "Viewing Software Information"
- Section 8.3.4, "Commands"
- Section 8.3.5, "Queue"
- Section 8.3.6, "Command History"
- Section 8.3.7, "Viewing Device Logs"

8.3.1 Viewing Device Information

The Mobile Manager displays general and database information for a chosen device. To view device information, click **Device Info**. If no information is displayed, click **Retrieve Device Information**. This sends a command to the device, which is then posted back to this page. Click reload until the information is posted.

In the first section, all of the details about the operating system is provided. You no longer have to go to the machine and type in a command to determine the operating system, its version and the latest service pack applied. This section will provide you with all of this information. In addition, you can see what the host name and IP address is.

The second section details how much memory you have on the device. This includes how much virtual or physical memory are on the device, and how much of that memory is still available.

The third section details the type of processor that is installed on the machine. For example, it describes the type of Intel processor that is installed on your Windows machine. You know exactly when your users must be upgraded to the next version of processor for the capability that they need.

For Windows-based devices only, the fourth section details the version of the JDK that you have installed and where it is installed. You no longer have to ask your users to check which version of JDK that they have installed. For example, in the Oracle Database Lite 10*g*, if you are using Oracle Application Server (9.0.4), you must be using the JDK 1.4.2. You can view this information for each device and know if the Mobile client software must be upgraded or not. In addition, this section describes the `CLASSPATH` for the Mobile client environment.

The last section details the amount of storage space that exists and is currently available on each drive.

8.3.2 Viewing Database Information

When you select the Database Info tab, you see all of the information about the Oracle Lite database that is installed on the Mobile client, which uses ODBC.

The first section provides the ODBC driver name, so that you can know which version that is installed on your client. In addition, you can see what DLL is used for the database and the directory.

The second section details each of the user ODB files—that is, the client databases for each application. To validate the file data integrity of the ODB file, select the button next to the ODB file and click **Validate**.

The third section displays the configuration in raw form in the `POLITE.INI` file on the client. Each section in the `POLITE.INI` file is displayed. The purpose of this section is for you to view the sections and parameters in the INI file. The data is shown in raw format.

8.3.3 Viewing Software Information

You can view all of the Oracle Database Lite software that is installed on the Mobile device by clicking **Applications**, which lists each application, its version, setup time, and location details. If no information is available, click **Retrieve Software Information**, which sends a command to the device.

Once the information becomes available, which is dependent on when the device reconnects, the platform is listed for the device. Select the platform to see the software information.

8.3.4 Commands

You can send commands to each device by itself or to all devices in a platform to gather information or execute some function. This is described fully in Section 8.6, "Sending Commands to Your Mobile Devices".

8.3.5 Queue

The queue shows any commands that are currently in process. That is, if the device is not currently online, then the command is placed into the queue until the device becomes available. Viewing this queue shows you all of the commands that are queued up waiting for devices.

8.3.6 Command History

This shows all of the commands executed against this device.

8.3.7 Viewing Device Logs

The Mobile Manager displays device logs and synchronization logs from the client device. To view the client device logs, click **Device Logs**. The Device Logs page lists what activity has occurred on the device. When you click Purge, these logs are removed. To view client device synchronization logs, click **Synchronization**. This shows only the synchronization requests made by the client.

8.4 Configuring and Customizing Your Mobile Device Platform

As described in Section 2.2, "Installing the Mobile Client Software", you normally install the Mobile client software by choosing the type of Mobile device and the language that you will use. However, you can extend any of these platforms to not only install the Mobile client software for the platform, but also install any of your binaries or applications. The following sections describe how to configure your platform or create and customize a new platform.

- Section 8.4.1, "Modifying Platform Properties for Installation"
- Section 8.4.2, "Enabling or Disabling All Mobile Devices in a Platform"
- Section 8.4.3, "Allowing Software Upgrades to All Mobile Devices in a Platform"
- Section 8.4.4, "Extend or Create a Custom Platform"

8.4.1 Modifying Platform Properties for Installation

Before you install a platform on your system, you should ensure that all of the configuration details for the Mobile device setup are what you want. A setup file is used to detail installation details, such as directories, binaries to install, registry to modify, path and `CLASSPATH` additions, and so on. You can modify the setup INF file that is defaulted for each platform, or you can create your own and point the platform to the new setup INF file.

Generally, you do not want to modify the generic platforms provided for you, in case you need to go back to basics. Thus, you should create your own unique platform by extending one of the provided platforms. This copies the existing platform into a separate platform with your name. Once copied, or extended, modify this platform with your own unique characteristics. For instructions on how to extend one of the provided platforms, see Section 8.4.4, "Extend or Create a Custom Platform".

To modify how the platform is installed, do the following:

1. Designate the name and path of the setup INF file for your platform by navigating to the Mobile Devices page.

   a. Click **Platforms**.

   b. Click on the platform for which you are currently modifying the INF file.

   c. Make sure that the correct setup INF file is listed in the Setup INF field.

   Note: If you want to modify this INF file or provide a different INF file, then on the Mobile Server, navigate on the file system to *`ORACLE_HOME`*`/Mobile/Server/admin/repository/setup/dmc/common` to where the setup INF files are located. Open the file that you want to modify or create a new INF file. Add the changes you want for this platform. See Section 8.10, "Installation Configuration (INF) File" for the configuration syntax.

   ---

2. Choose a Bootstrap group command from the list displayed. After the device bootstrap is complete, you can choose a group command to execute when it is completed. For example, choosing the device information command retrieves all of the device information to the Mobile Manager for viewing.

3. Enable or disable all devices in the platform. If you enable these devices by choosing Yes, then each user can log in, perform updates, synchronize, and perform other duties. If you disable these devices by selecting No, then they can no longer perform work for the user. See Section 8.4.2, "Enabling or Disabling All Mobile Devices in a Platform" for more information.

4. Set Upgradable to Yes to retrieve all software updates for all devices in the platform. If you want these devices to continue to receive automatic software updates, choose Yes. If you want these devices to stay with the current software versions, choose No. See Section 8.4.3, "Allowing Software Upgrades to All Mobile Devices in a Platform" for more information.

8.4.2 Enabling or Disabling All Mobile Devices in a Platform

You can enable or disable all Mobile devices in a platform. By default, each device in the platform is enabled, which means that the user can synchronize to the database and perform software updates. If you disable the device, then it can no longer perform work for the user. If you wanted to disable a single device—because a user has lost the device or left the company—then you follow the instructions in Section 8.5.1, "Enabling or Disabling a Mobile Device". However, if you want to enable or disable all devices for a platform, then see Section 8.4.1, "Modifying Platform Properties for Installation".

Why would you want to disable all devices in the platform? What if you had created a a customized platform (see Section 8.4.4, "Extend or Create a Custom Platform") for devices that were used for a specific purpose, such as if you had cell phones that were analog only. When you came out with a full digital network, you may not want any of the analog technology to continue to be used. You could choose to send a deinstall command (see Section 8.6, "Sending Commands to Your Mobile Devices") and then disable all of the analog Mobile devices. Since all of them had the same platform, all of them could be disabled at the same time. The user could no longer log in and use the device. They would be forced to upgrade to digital.

8.4.3 Allowing Software Upgrades to All Mobile Devices in a Platform

If you set Upgradable to Yes, then when a new software update comes available—either for the Mobile client software or for any applications installed on the client—then the Mobile device will receive these updates. However, if you want a device to stay with the level of software that is currently installed, then you would change Upgradable to No. See Section 8.5, "Configuring Your Mobile Devices" for where you set whether the software is to be upgraded or not.

8.4.4 Extend or Create a Custom Platform

You can create custom device platforms either by using an existing platform as the template or by enabling a platform.

- Only a few of the available platforms are displayed in the Mobile client setup screen. To add a platform that you need, enable the desired platform. See Section 8.4.4.1, "Enable a Platform for Your Mobile Client" for more information.
- You can create a new platform by extending one of the existing platforms. This enables you to customize the platform to install additional binaries, applications, or to have specific instructions on modifying the client machine to accommodate your specifications. See Section 8.4.4.2, "Create a Custom Platform By Extending an Existing Platform" for more information.

8.4.4.1 Enable a Platform for Your Mobile Client

Not all of the possible platforms are enabled on the Mobile client setup screen. To enable a platform for your client device, do the following:

1. On the Mobile Devices screen, click **Platforms**.
2. On the Platforms screen in the Search pulldowns, select the language and either Disabled or All and click **Go**.
3. Select the platform name that you want to enable.
4. Enable the device by selecting Yes in the Enable pulldown.
5. Click **OK**. The device is now enabled and will be visible in the client setup screen.

8.4.4.2 Create a Custom Platform By Extending an Existing Platform

To create a custom platform, do the following:

1. Create a new INF file for your extended platform—On the Mobile Server, navigate on the file system to `*ORACLE_HOME*/Mobile/Server/admin/`

repository/setup/dmc/common to where the setup INF files are located. Create an empty INF file for your new platform. As discussed in Section 8.10, "Installation Configuration (INF) File", when you extend a platform, the INF files that are used for the installation is a concatenation of your platform and every platform that it was extended from—just like how objects extend methods, properties and attributes from each other in an object-oriented language.

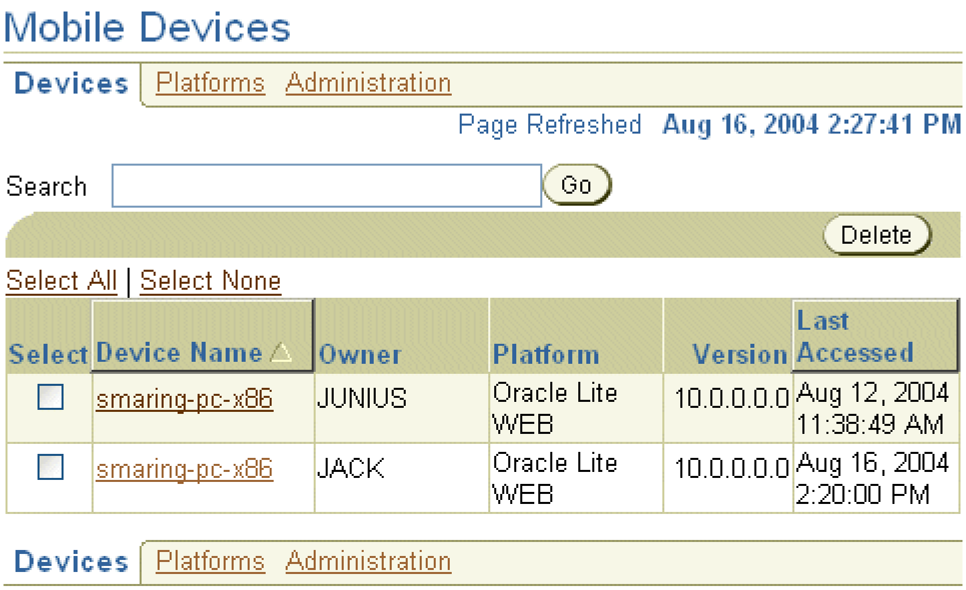
2. On the Mobile Devices screen, click **Platforms**.
3. On the Platforms screen, select the platform name and click **Extend**.
4. Enter the custom platform name, path, and file name of the blank setup INF file you created in step 1. The setup INF file determines what is installed and how the client machine environment is modified. See Section 8.10, "Installation Configuration (INF) File" for instructions on how to modify the setup INF file after you have completed extending the platform.
5. Choose a Bootstrap group command from the list displayed. After the device bootstrap is complete, you can choose a group command to execute when it is completed. For example, choosing the device information command retrieves all of the device information to the Mobile Manager for viewing.
6. Enable or disable the device. If you enable the device by choosing Yes, then the user can log in, perform updates, synchronize, and perform other duties. If you disable the device by selecting No, then it can no longer perform work for the user.
7. Set Upgradable to Yes to retrieve all software updates for the device. If you want the device to continue to receive automatic software updates for the device, choose Yes. If you want the device to stay with the current software versions, choose No.
8. Click **OK**.
9. After you have extended your platform and given it a unique name, you should modify the setup INF file for this platform.

You are ready to install your customized platform.

8.5 Configuring Your Mobile Devices

Navigate to the Mobile Devices page, as shown in Figure 8–4, and you can modify, delete, and extend any Mobile device.

Figure 8–4 Devices Page



Select	Device Name △	Owner	Platform	Version	Last Accessed
☐	smaring-pc-x86	JUNIUS	Oracle Lite WEB	10.0.0.0	Aug 12, 2004 11:38:49 AM
☐	smaring-pc-x86	JACK	Oracle Lite WEB	10.0.0.0	Aug 16, 2004 2:20:00 PM

To modify the configuration of a specific device, select the device and do the following:

1. The device name is displayed on the first line. You can modify the name of the device to anything that you want. For example, if you have a naming convention for all devices in your organization, modify this field to reflect this convention. The name defaults to the Mobile device platform.
2. Enable the device by selecting Yes; disable the device by selecting No. See Section 8.5.1, "Enabling or Disabling a Mobile Device" for more information.
3. Set Upgradable to Yes to retrieve all software updates for the device. If you want the device to continue to receive automatic software updates for the device, choose Yes. If you want the device to stay with the current software versions, choose No. See Section 8.5.2, "Allowing Software Upgrades to the Mobile Device" for more information.
4. The installed address is displayed in the Address field. If the address is an IP address or a phone number, these may change at some point. You can enter the new addresses in this field.
5. Choose the Network Provider type. You can choose a different network provider than that with which you chose to install. The default list includes HTTP, WOR_IAS (Wake on Ring with Oracle Application Server), WOR_SMTP (Wake on Ring with SMTP), SMS, or RAPI. If you added another network provider, these custom network providers will also be included in the list.

Note: To create a network provider, see Section 8.9, "Managing the Network Protocol Between the Device and the Mobile Client Software".

8.5.1 Enabling or Disabling a Mobile Device

You can enable or disable a Mobile device. By default, the device is enabled, which means that the user can log in, perform updates, synchronize, and perform other duties. If you disable the device, then it can no longer perform work for the user. See

Section 8.5, "Configuring Your Mobile Devices" for where you enable or disable the device.

Why would you want to disable the device? If the Mobile device was lost or the user is no longer with the company, but did not return the device, then you might choose to send a deinstall command (see Section 8.6, "Sending Commands to Your Mobile Devices") and then disable the device. This way, the software is no longer on the device and even if the user had another copy of the software to reinstall the application, they could no longer log in and retrieve any information from your company.

8.5.2 Allowing Software Upgrades to the Mobile Device

If you set Upgradable to Yes, then when a new software update comes available—either for the Mobile client software or for any applications installed on the client—then the Mobile device will receive these updates. However, if you want a device to stay with the level of software that is currently installed, then you would change Upgradable to No. See Section 8.5, "Configuring Your Mobile Devices" for where you set whether the software is to be upgraded or not.

8.6 Sending Commands to Your Mobile Devices

As an administrator, you can create and send commands to any Mobile device. These commands can do a range of functions. The following sections describe how to send existing commands to devices and how to create new commands for your own purposes:

- Section 8.6.1, "Scheduling or Sending Commands"
- Section 8.6.2, "Modifying Existing Commands"
- Section 8.6.3, "Creating New Commands"
- Section 8.6.4, "Creating Group Commands"
- Section 8.6.5, "Enabling or Disabling Mobile Device Commands"
- Section 8.6.6, "Viewing the Mobile Device Command History"

8.6.1 Scheduling or Sending Commands

You can send or schedule a command to be sent from the Mobile Manager. Navigate to the Mobile Devices page.

8.6.1.1 Sending Commands

You can send commands to devices that are installed and registered with the Mobile Manager. You can send these commands from several places.

- Sending a command to a single device—To send a command to a single device, select the device from the list displayed on the Mobile Devices page. Select **Commands**. Under the Send Commands section, choose the command—as designated by the description—and click **Send Now**. The Mobile Manager seeks your confirmation and then displays a confirmation message.

  If the command requires arguments, then the Mobile Manager displays an argument collection page. For example, the Upload File command requires a file name as an argument. To send the command to the device, click **Yes**.

- Sending a command to all devices of the same platform type—To send a command to all devices of a certain platform, click **Platform** off of the Mobile Devices page. Click the select button next to the desired platform, select the command from the command pull-down list, and click **Send Command**.

For information on how to create a command, see Section 8.6.3, "Creating New Commands".

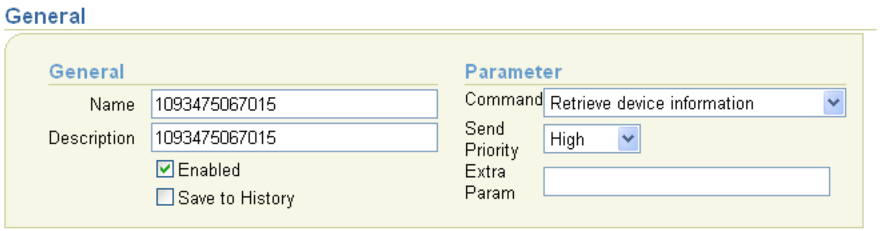
8.6.1.2 Scheduling Commands

You can schedule a command to execute at a later time or at certain intervals. Select the device to which you want this command to be directed. Select **Commands**. Perform the following:

1. Click **Schedule**. The Schedule Command page appears.
2. As shown in Figure 8–5, configure the following:
   - The name and descriptor are unique identifiers. You can modify them to your own unique identifiers.
   - Choose the command that you want to schedule from the Parameter Command pull-down list.
   - Check the Enabled box to enable or disable the command. If disabled, the command cannot be executed.
   - Check Save to History if you want to keep a log of when this is executed and the results, which are printed to the Command History screen.
   - Choose from the Priority pull-down list if this is to be high, medium, or low priority. This determines in what order the scheduled commands are executed.
   - Enter any expected parameter values, separated by semi-colons, in the Extra Parameter field. For example, if you chose the Synchronize databases command and you wanted a fast refresh, you would enter 'fast' in the Extra parameter field.

   > **Note:** You cannot use fast refresh synchronization with high priority restricting predicate.

Figure 8–5 The General Section of the Command Scheduler



General

General

Name: 1093475067015

Description: 1093475067015

☑ Enabled

□ Save to History

Parameter

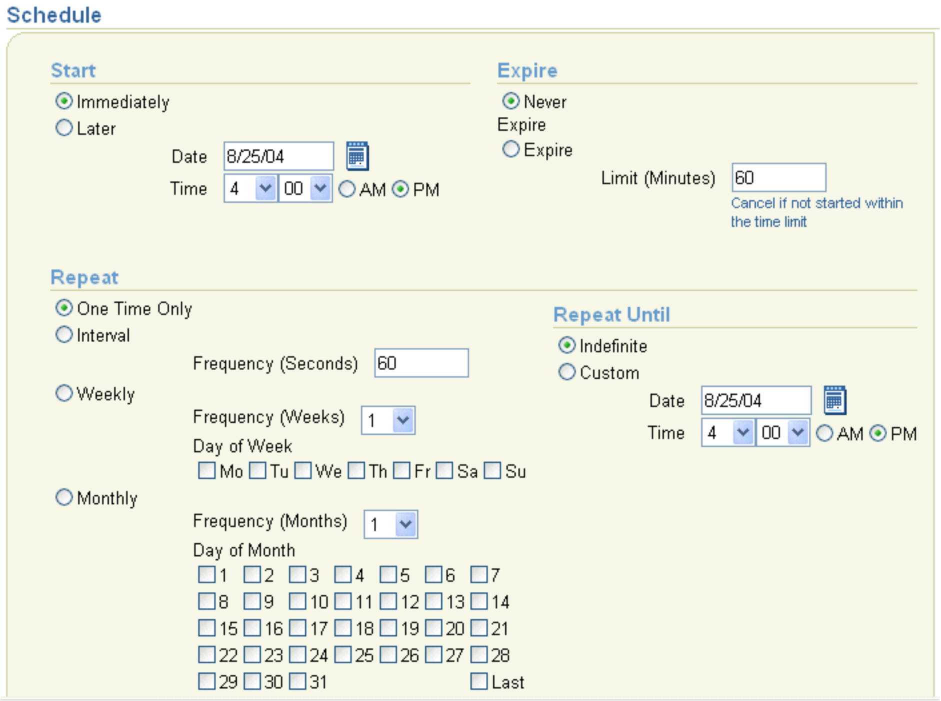
Command: Retrieve device information

Send Priority: High

Extra Param:

3. As Figure 8–6 shows, enter the timing that the command is to execute. Choose when it is to start, when it will expire (if it does not execute within a certain time frame), how often it will repeat, and a date and time that it will repeat until.

Figure 8–6 Timing Section of the Command Scheduler



Schedule

Start

● Immediately

○ Later

Date 8/25/04

Time 4 00 ○ AM ● PM

Expire

● Never Expire

○ Expire

Limit (Minutes) 60

Cancel if not started within the time limit

Repeat

● One Time Only

○ Interval

Frequency (Seconds) 60

○ Weekly

Frequency (Weeks) 1

Day of Week

□ Mo □ Tu □ We □ Th □ Fr □ Sa □ Su

○ Monthly

Frequency (Months) 1

Day of Month

□ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
□ 8 □ 9 □ 10 □ 11 □ 12 □ 13 □ 14
□ 15 □ 16 □ 17 □ 18 □ 19 □ 20 □ 21
□ 22 □ 23 □ 24 □ 25 □ 26 □ 27 □ 28
□ 29 □ 30 □ 31 □ Last

Repeat Until

● Indefinite

○ Custom

Date 8/25/04

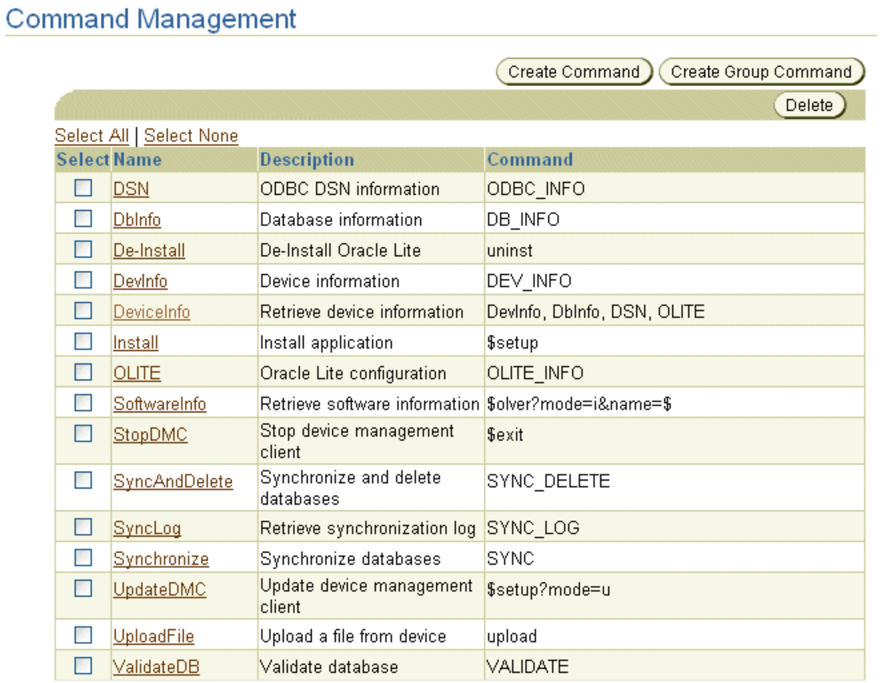
Time 4 00 ○ AM ● PM

4. Click **Apply**. The Mobile Manager displays a confirmation message.

8.6.2 Modifying Existing Commands

To view the available commands, select the Mobile Devices tab on the Mobile Manager. Navigate to the Administration page and click the Command Management link. As Figure 8–7 displays, the Command Management page appears.

Figure 8–7 Command Management Page



Command Management

Select	Name	Description	Command
☐	DSN	ODBC DSN information	ODBC_INFO
☐	DbInfo	Database information	DB_INFO
☐	De-Install	De-Install Oracle Lite	uninst
☐	DevInfo	Device information	DEV_INFO
☐	DeviceInfo	Retrieve device information	DevInfo, DbInfo, DSN, OLITE
☐	Install	Install application	$setup
☐	OLITE	Oracle Lite configuration	OLITE_INFO
☐	SoftwareInfo	Retrieve software information	$olver?mode=i&name=$
☐	StopDMC	Stop device management client	$exit
☐	SyncAndDelete	Synchronize and delete databases	SYNC_DELETE
☐	SyncLog	Retrieve synchronization log	SYNC_LOG
☐	Synchronize	Synchronize databases	SYNC
☐	UpdateDMC	Update device management client	$setup?mode=u
☐	UploadFile	Upload a file from device	upload
☐	ValidateDB	Validate database	VALIDATE

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

1. Click the required Command Name link. The Properties page for this command appears.
2. Enter the command name, description, and syntax in the corresponding fields. For more information on modifying these fields, see Section 8.6.3, "Creating New Commands".
3. To check the accuracy of the command syntax, click **Syntax Check** button. If no errors are found, the Mobile Manager displays a confirmation message.
4. Click **Apply**. The Mobile Manager displays a confirmation message.

8.6.2.1 Adding Parameters to Mobile Device Commands

You must configure the command to prompt for expected input parameters. For example, the Synchronize command requires that you define what type of refresh you want: fast, force, or push.

You can specify any parameters by modifying the command, as follows:

1. Click the required Command Name link. The Properties page for this command appears.
2. Add the parameter name and values, as follows:
   - The parameter name—This is the name specified in the OTL script.
   - A short description—The description is what is displayed when the user is prompted for the value of the parameter.
   - The display name—This is the description for each value. For example, the Synchronize databases command has three possible values: fast, force or push. However, the display values to describe each of these actual values is Fast

Refresh, Force Refresh, and Push Only. These values are separated by a semi-colon, as follows: Fast Refresh;Force Refresh;Push Only.

> **Note:** You cannot use fast refresh synchronization with high priority restricting predicate.

- The default values for this parameter—Enter one or more potential values for this parameter, if applicable. For example, the Synchronize databases command values would be fast;force;push. These values are separated by semi-colons and in the same order as the display name. If you do not have definitive values, leave blank and the user will enter their own value.

8.6.3 Creating New Commands

You can create commands using the OTL scripting language, as described in Section 8.11, "Defining Device Manager Commands With the Device Manager OTL Tag Language". These commands are then used to perform activity on the Mobile devices, but controlled by the administrator within the Mobile Manager.

You can create a command with a single or multiple OTL script commands. Each is created in a different manner, as described in the following sections:

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

To create a command that has several lines, you must perform the following:

1. Create a file with an `.otl` extension with the OTL commands in it. Place this file in the *`ORACLE_HOME`*`mobile_oc4j\j2ee\mobileserver\applications\mobileserver\setup\dmc\otl` directory.
2. With any editor, add all OTL commands that you want executed within the file. See Section 8.11, "Defining Device Manager Commands With the Device Manager OTL Tag Language" for a full description of the OTL scripting language.
3. Within Mobile Manager, navigate to the Command Management page.
4. Click **Create Command**.
5. In the Name field, pick a short name to identify the command within Mobile Manager.
6. In the Command field, put the name of the OTL file, without the `.otl` extension.
7. In the Description field, type in a sentence describing accurately the purpose of the command.
8. Click **OK**.
9. Back on the Command Management screen, select the command that you just created.
10. Click **Syntax Check** to ensure that you entered in your OTL commands correctly.
11. Click **Apply**.
12. If you ask the user to enter parameters, then add the parameter definitions. See Section 8.6.2.1, "Adding Parameters to Mobile Device Commands" for a full description.
13. Push the OTL script to device.

You have now successfully created a new command. You can now execute this command against your Mobile devices. See Section 8.6.1, "Scheduling or Sending Commands" for information on how to execute the command.

8.6.4 Creating Group Commands

To create group commands, do the following:

1. Click **Create Group**. The Create Group Command page appears.
2. Enter a unique Command Name, which will be used to identify the grouping.
3. Enter a description.
4. Select the set of existing commands that you want to execute together. The Command Weight feature controls the order in which the commands are 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.
5. Click **Add**. The Mobile Manager displays a confirmation message.

8.6.5 Enabling or Disabling Mobile Device Commands

By default, all commands are enabled, which means that you can execute the command. If you want to disable a command, so that it can no longer be executed, do the following:

1. From the Mobile Devices page, click **Administration**.
2. Choose **Command Management**.
3. Select the command that you want to enable or disable.
4. Select either Yes for enable or No for disable on the Enabled pull-down.
5. Click **Apply**.

8.6.6 Viewing the Mobile Device Command History

To view the Device Command History, click the Command History link from the single Mobile device screen. The Command History page lists a history of commands that were implemented for the chosen device. You can delete a single historical message by clicking the select box next to the message and clicking **Delete**. To delete all messages, click **Purge**.

8.6.7 Examples of Mobile Commands

The following are examples of commands that you can send to your Mobile device:

- To create a command that configures for automatic encryption of a local Oracle Lite database, see Section 6.4.2, "Configuring on the Server for Automatic Encryption of Local Snapshots".
- To trigger a synchronization on the client, see Section 2.2.1, "Configuring for Default Sync When Installing the Client".

8.7 Enabling Device Software Updates

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

- Section 8.7.1, "Enabling Major Software Updates for Your Device"
- Section 8.7.2, "Applying Patches or Minor Updates"
- Section 8.7.3, "Controlling Device Software Updates"

8.7.1 Enabling Major Software Updates for Your Device

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

To update your software, perform the following:

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

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

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

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

8.7.2 Applying Patches or Minor Updates

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

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

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

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

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

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

```
<setup name="Application Name" version="1.2.3" id='1001'>
   <install>
     <action msg_i='$FILE_I$' msg_u='$FILE_U$'>file</action>
       <file>
         <item>
           <src>/common/win32/olobj40.dll</src>
           <des>$APP_DIR$\bin\olobj40.dll</des>
         </item>
       </file>
   </install>
</setup>
```

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

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

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

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

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

Update the patch, as follows:

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

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

8.7.3 Controlling Device Software Updates

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

- Section 8.7.3.1, "Enabling Software Updates for the Oracle Lite Platform"
- Section 8.7.3.2, "Updating Application Software On Each Client"

8.7.3.1 Enabling Software Updates for the Oracle Lite Platform

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

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

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

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

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

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

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

8.7.3.2 Updating Application Software On Each Client

You can control whether a new version of an application software is downloaded on each client. Modify the update policy attribute of the user in one of the following ways:

- On the user page in the Mobile Manager, set the Software Update pulldown to the appropriate update that you want, as follows:
  - All updates—Include major and minor updates.
  - Major—The devices attached to this user receives only major software updates. This is the default.
  - Minor—The devices attached to this user receives only minor software updates.
  - Disable updates—The devices attached to this user does not receive any software updates.

  In addition, you can specify the date that the update occurs.
- Set the `UPDATE_SOFTWARE_APPS` policy attribute of the `User` object to one of the following values to specify what type of update that the client can receive:

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

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

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

Example 8–1 Upgrading Devices Attached to a Specific User

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

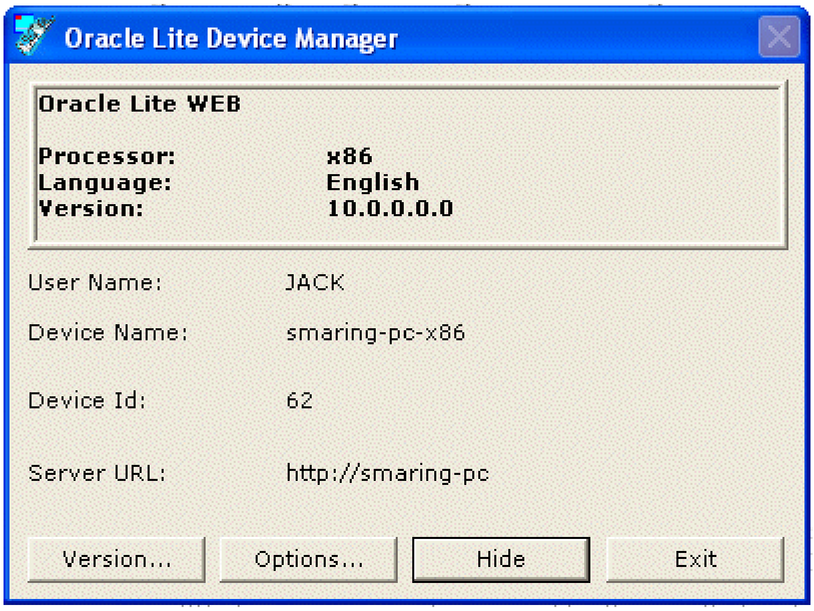
For example, you have two users: John and Tom. You want John's devices to stay at the current version, which is Oracle Database Lite Win32 version 10.0.0.0.0; however, you want Tom's devices to upgrade to the new version, which is Oracle Database Lite Win32 version 10.1.0.0.0. Configure each user's devices,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

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

8.8 Using the Device Management Client GUI to Manage Device on Client-side

On any client—except for the Palm clients—you can manage the Mobile device client software, as shown in Figure 8–8.

Figure 8–8 Using the Oracle Lite Device Manager to Manage Your Device



Oracle Lite WEB

Processor: x86
Language: English
Version: 10.0.0.0.0

User Name: JACK

Device Name: smaring-pc-x86

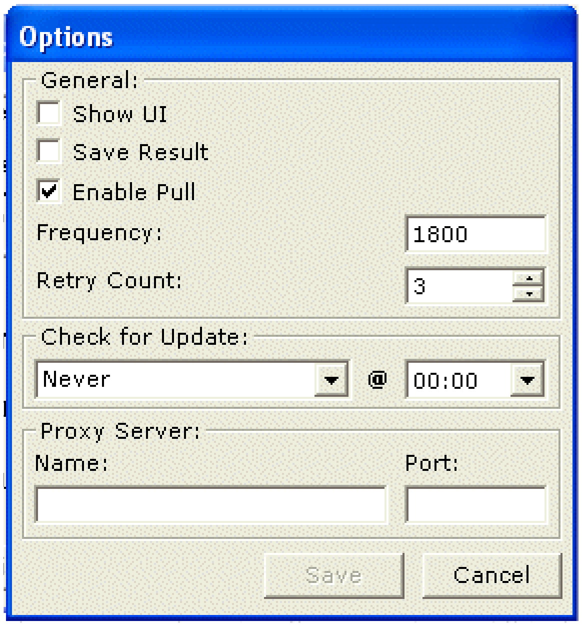
Device Id: 62

Server URL: http://smaring-pc

To bring up the Device Manager GUI, choose **Oracle Lite Device Manager** from the Oracle Database Lite Programs list. The main screen provides information about the following: platform with type of platform, language, and version, the owner/user of this device, the name of the device, and the URL of where the device is installed.

- Click **Version** to see the version number of all Oracle Database Lite software DLLs.
- Click **Options**, which brings up Figure 8–9, to configure the following:
  - Enable Pull: If the server cannot connect to the client, any commands sent to the client are placed into the Device Manager command queue. These commands are sent to the client when one of two things occurs:
    - The client synchronizes.
    - A "Pull" is initiated from the client. When you check the **Enable Pull** checkbox, the device management client automatically polls the Device command queue for any commands for this user. The frequency is the number of seconds to wait between each pull.
  - Retry Count: If you have created your own Device Command, this count specifies the number of times that the command is executed, if it fails to execute. If it still fails after retrying, the command is deleted. This count also applies to failed synchronization attempts. The commands are retried with the same frequency interval that is set for the Enable Pull command. A command can fail to execute if there is an error within the command or if there is no connection between the client and the server.
  - Check for Update: Select the day and time that you want the device management client to automatically poll for updates. This also occurs anytime you start a synchronization. So, this is useful if you never synchronize with the Mobile Server.
  - Proxy Server: If you have a proxy server between the Mobile client and Mobile Server, enter the address and port here.

Figure 8–9 Device Client Manager Options



☐ Show UI
☐ Save Result
☑ Enable Pull
Frequency: 1800
Retry Count: 3
Never @ 00:00
Name:
Port:

- Click **Hide** to place the **Oracle Lite Device Manager** in the Windows System Tray.

8.9 Managing the Network Protocol Between the Device and the Mobile Client Software

The Network Management page is where the administrator defines the properties of an installed network provider or register new network providers. A network provider is the protocol that the Mobile client uses to communicate between itself and the Mobile device. The Mobile client software, which is often installed on a Windows system, sends commands to the Mobile device over this protocol. Often, you have a device, such as WinCE or Palm, that interacts with the Mobile client installed on a Windows system.

This network provider definition describes what you have already installed as the protocol between the Mobile client and the device. The frequently-used network providers are as follows:

- HTTP—If you use HTTP, you will provide an HTTP URL in the Address field. You cannot use HTTPS between the Device Manager and the Mobile device.

  > **Note:** HTTP may not work if the device does not have a direct IP conection to the Mobile Server.

- RAPI—Remote API used by the ActiveSync API, which only supports Pocket PC class devices. These devices connect directly to the computer that is executing the Mobile Server.
  - RAPI does not work on LINUX or UNIX-based systems.
  - When using RAPI on a Windows machine, install and configure ActiveSync 3.7.1.
  - Set `CEUTIL.DLL` and `RAPI.DLL` in the `%WINDIR%\System32` directory.
  - In addition, if your Mobile Server uses Oracle Application Server as its middle-tier solution, set `%WINDIR%\System32` in the application server path.
- Wake on Ring—If you have a cell phone as a Mobile device, then you would have a network protocol where the cell phone receives incoming data. Thus, the address is a phone number for the cell phone. The cell phone is "woken" when incoming commands are initiated from the Mobile Manager. There are two types of Wake on Ring:
  - WOR_SMTP: Wake on Ring over SMTP.
  - WOR_IAS: Wake on Ring over Oracle Application Server.
- SMS—Short Messaging Service.

To modify or create a new network provider, navigate to the Devices page, click **Administration**, and click **Network Management**. The Network Management page lists existing network providers. Select any of these providers to see their properties, which consists of the following:

- Java class name: The name of the JAVA class that implements the Network protocol, such as HTTP.
- Meta data: Any user defined string that is required as input by the JAVA class during the initialization.

To define a new network protocol, do the following:

1. Create a Network Provider class using Java. This class must implement the `oracle.lite.provider.NetworkProvider` interface.
2. Register the Network Provider through the Mobile Manager on the Network Management page, as follows:
    a. Click **Create** on the Network Management page.
    b. Input the network provider name, Java class name, and Meta data.
    c. Click **OK**.

8.10 Installation Configuration (INF) File

The Installation Configuration file contains all the instructions required to install or de-install client software and its format is based on XML. The INF file contains a set of actions and each action may have multiple items.

When you extend a platform, the INF files that are used for the installation is a concatenation of your platform and every platform that it was extended from—just like how objects extend methods, properties and attributes from each other in an object-oriented language. For example, the Branch Office INF file is extended from the Web-to-Go INF file. Thus, when you install Branch Office, the Mobile Server concatenates instructions for the installation from both the Branch Office and Web-to-Go INF files. Any modifications added to the Web-to-Go INF file since the extension will still apply to the installation as both INF files are read at installation time. When you view the configuration information on Mobile Manager, the type field describes all of the platforms that this INF file extends from.

As Table 8–1 describes, the supported keywords start with a '$' character and ends with a '$' symbol.

Table 8–1 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, which can be one of the following: WIN32, WINCE, LINUX, PALM.
`$OS_VER$`	Operating system version. For example, NT, 95, XP, 3.0, and so on.
`$OS_LANG$`	Language or Location name, which can be US for English or JA 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 URL, if any.
`$SERVER_URL$`	Oracle Mobile Server URL.

8.10.1 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">
<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, which can only be cab to specify a Windows CAB format.

8.10.2 Properties

All of 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.

  `<storage>5</storage>`

- MEMORY - Minimum amount of system memory in MB. Required.

  `<memory>5</memory>`

- LOCATION - Location or directory name of the application. You can specify the location for the application in one of the following ways:

  – Default directory. If you do not specify anything or specify `<location></location>`, then the directory defaults to mobileclient, which is created on the drive with the largest available free space. The user is always prompted to either accept the default or enter the directory that they wish.

  – Absolute directory. Define the absolute path where the application is to be installed. The following installs the application in the c:\abc directory:

  ```
  <location>c:\abc</location>
  ```

  – Specify a default directory name. The directory will be created on the drive with the largest available free space. In addition, the user is always prompted to be able to alter the directory into which the application is installed. The following example defines the default directory as abc:

  ```
  <location default='abc'></location>
  ```

  – Define a default directory and an absolute directory. You can specify an absolute directory, where the drive may not exist. If the drive does not exist, then a prompt appears with the default directory, where the user can accept he default or provide another. The following defines the absolute directory of e:\abc, which if the E drive does not exist, then the default directory of abc is created on the drive with the most available free space:

```
<location default=’abc’>e:\abc</location>
```

- Specify the platform(s) that this application is installed upon. You can define, with the type attribute, what platforms this application is to be installed on or not installed on. The platforms that you can specify are WIN32, LINUX, and WINCE.

  - To install only on WIN32, do the following:

    ```
    <location default=’abc’ type=’WIN32’></location>
    ```

  - To install on all platforms, except WIN32, do the following:

    ```
    <location default=’abc’ type=!’WIN32’></location>
    ```

  - To install on either WIN32 or WINCE, do the following:

    ```
    <location default=’abc’ type=’WIN32|WINCE’></location>
    ```

- `PROMPT` - You can have a window pop-up with a prompt if one of the files that you need to install is currently being used. If the user inputs Yes, then the other instances using that file are terminated. For example, if other executables are using the `ologj40.dll` when you are installing the client, the prompt "Would you like to terminate the Oracle Lite Application?" is provided to the user.

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

  The following is an example of the `setup` section in the INF file:

  ```
  <setup name="Oracle Lite" version="1.0.0.0">
  <property>
          <storage>4</storage>
          <memory>12</memory>
          <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>
  ```

8.10.3 Initialization

Initialization includes setting keywords that you can use when installing your application; the Oracle Database Lite installation keywords are described in Table 8–1. Specify a keyword for your application installation in the `type` parameter and its value in the `name` parameter. The following defines a `WIN32` keyword with a value of `$APP_DIR$/bin`.

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

8.10.4 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.

8.10.5 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 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. If you have a child element without a corresponding action element, it will not be executed. The action elements force the invocation of the child elements.

Table 8–2 describes INSTALL actions that are supported by the SMC.

Table 8–2 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.
register	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 element contains names of all the directories to be created during the installation process. Entries in this section are fully qualified directory names. For example,

```
<directory>
    <item>$APP_DIR$\oldb40</item>
    <item>$APP_DIR$\crm</item>
</directory>
```

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

FILE

The file element lists all the files to be copied during the software installation process. Each item contains a target file name and source file name. The source file name must be unique. We do not support copying the same source file to multiple destination files. For example,

```
<file>
```

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

If you want to copy a source file multiple times, you cannot just define the source file and then configure for it to be copied to multiple destinations. Instead, you must manually copy the source file to another filename and then configure it as follows:

```
<file>
  <item>
        <src> win32/crm/crm.dll </src>
        <des>$APP_DIR$\crm\crm1.dll </des>
   </item>
   <item>
        <src> win32/crm/crm2.dll </src>
        <des>$APP_DIR$\crm\crm2.dll </des>
   </item>
</file>
```

Where `crm.dll` and `crm2.dll` are the same source file.

ENV

The `ENV` element contains all environment variables to be added to a Windows NT registry. This modifies only the User environment in Windows NT systems.

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

The above example appends the `application_root\webtogo` directory to the `PATH` environment variable.

REGISTRY

The `registry` element 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>
```

The SMC adds the Windows Registry key named `Test` in the directory named `HKEY_CURRENT_USER\Software\Oracle` 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

The JAVA element 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, which is downloaded from the Mobile Server.

```
<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

The LINK element creates a symbolic link on the client system, such as a UNIX soft link or a Windows program link (or Program menu item). Each entry must have a name, a program file name and a folder name, which describe where you want to put the symbolic link and the file path.

The following example creates a symbolic link on a UNIX platform to libolite40.so.1 in $ADD_DIR/bin directory, which points to APP_DIR/bin/libolite40.so.

```
<link>
    <item name='libolite40.so.1'>
        <folder>$ADD_DIR/bin</folder>
        <file>$APP_DIR$/bin/libolite40.so/file>
    </item>
</link>
```

For Windows platforms, you can also optionally set the current working directory with the <directory> tag and the default arguments can be set using the <arg> tag. The following example creates a Windows program link to Oracle Web-to-Go.lnk, where the .lnk is automatically appended, in the Startup folder for the webtogo.exe file, which is located in the $APP_DIR\bin directory.

```
<link>
    <item name='Oracle Web-to-Go'>
        <folder>Startup</folder>
        <file>$APP_DIR$\bin\webtogo.exe</file>
        <directory>$APP_DIR\bin</directory>
    </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="DATABASE_ID">200</item>
            <item name="NLS_LANGUAGE">ENGLISH</item>
    </item>
</ini>
```

If you want to replace an existing item in the INI file, just provide the name and value, such as follows:

```
<item name="DATA_DIRECTORY">c:\mobileclient</item>
```

The replace attribute defaults to true; thus `data_directory` is modified—whether it already exists or not in the INI file—to be `c:\mobileclient`. However, if you do not want to overwrite any existing value, but want only to add `DATA_DIRECTORY` if it does not exist, then set the `replace` attribute to false. The following example only adds `DATA_DIRECTORY` with `c:\mobileclient` if `DATA_DIRECTORY` is not currently configured in the INI file. If it is configured, the value is not replaced.

```
<item name="DATA_DIRECTORY' replace="false">c:\mobileclient</item>
```

The default value for the `replace` attribute is true; thus, if you want to replace the value, then set the `replace` attribute to false.

EXECUTE

The `EXECUTE` element 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 value defines how long the installer waits until it moves on to the next action. The wait value can be either an event name, multiple event names, or time specified in seconds. If the value is seconds, then the installer waits that many seconds and then moves on to the next action. However, if the wait value is an event name(s), then the installer waits for the executable (in this case, the `webtogo.exe`) to post that event, before the installer moves on to the next action. For example,

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

REGISTER

The `REGISTER` element 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>
```

8.11 Defining Device Manager Commands With the Device Manager OTL Tag Language

You can send a command from the Mobile Server to any device. To create these commands, use the Device Manager Tag Language, which is described in the following sections:

- Section 8.11.1, "Device Manager Tag Language Data Types"

- Section 8.11.2, "Operators That You Can Use With the Device Manager Tag Language"
- Section 8.11.3, "Syntax for the Device Manager Tag Language"
- Section 8.11.4, "Conditional Statements"
- Section 8.11.5, "Define Custom Functions"
- Section 8.11.6, "Manage the Database Connection"
- Section 8.11.7, "Global Classes"
- Section 8.11.8, "Importing Another OTL Page"
- Section 8.11.9, "Installing an Application From a Server"
- Section 8.11.10, "Specifying a Schema File"
- Section 8.11.11, "Error Handling"
- Section 8.11.12, "Sample Device Manager Commands Using the Tag Language"

8.11.1 Device Manager Tag Language Data Types

The allowed data types for the device manager are as follows:

8.11.1.1 Character

The Character object represents a UNICODE character. It is a primitive data type with no public methods. However, this data type supports implicit conversion methods, such as `toString()`.

8.11.1.2 Number

The Number data type represents either an integer, a double (float), or a large number.

8.11.1.3 Integer

The Integer object represents a four byte signed value. It is a primitive data type with no public methods.

8.11.1.4 Long

The Long object represents an eight byte signed value. It is a primitive data type with no public methods.

8.11.1.5 Double

The Double object represents a signed double (float) value. It is a primitive data type with no public methods.

8.11.1.6 Boolean

The Boolean object has only two possible values of `true` or `false`. It is a primitive data type with no public methods.

8.11.1.7 String

The String object represents a series of NULL terminated characters. The String data type represents all of the literal strings in OTL. They are immutable and has the following public methods:

Length ( )

Returns the number of characters in the string.

SubString (Integer start, Integer end)

Creates a sub-string from a String object. Provide the start and the end of the index and it returns the sub-string—beginning at the start value and stopping at the end value.

Trim ( )

Trim a string to remove white spaces from both ends of the string.

IndexOf (Character ch) or IndexOf (String str)

Find the index of a character or a substring within the string. Provide the character or substring inside the string to search for and the index of the first occurrence of the character or substring is returned.

LastIndexOf (Character ch) or LastIndexOf (String str)

Find the index of the last occurrence of a character or a substring within the string. Provide the character or substring inside the string to search for and the index of the last occurrence of the character or substring is returned.

EqualsIgnoreCase (String str)

Compares two strings, without comparing the case of the characters within the string. On one string, execute this method and provide the string to compare it to within the input parameter. True or false is returned.

StartsWith (String str)

Check if the string starts with the provided sub-string. True or false is returned.

EndsWith (String str)

Check if the string ends with the provided sub-string. True or false is returned.

ParseNumber ( )

Parse the string and create a number. This method succeeds only if the string represents a valid number. A number object is either an Integer, Double, or Long. For example, if the content of the String is ’12’, then this method returns an Integer of 12.

Replace (String in, String repl)

Replace a substring with another, as follows:

```
<c:set var='str' value='${str.Replace ("123,"345")}'/>
```

ToUpperCase ( )

Converts characters in the string to upper case.

ToLowerCase ( )

Converts characters in the string to lower case.

Tokenize (Character sep)

Tokenize the string into sub-strings, each separated by a character separator. The input parameter is the character that is to be used as the character separator. The output is an

Enumeration object. For example, the following OTL script separates numbers by separating a string everytime it encounters a semi-colon:

```
<c:set var="str" value="1;2;3;4"/>
<c:foreach var="tok" items="${str.Tokenize (';')}">
 Token = <c:out value="${r}"/>
</c:foreach>
```

8.11.1.8 Array

The OTL Array object can hold a set of other objects. An array can hold dissimilar objects and can grow automatically as more objects are added. All of the array objects have a global scope.

Sort (Boolean ascend)

Sort the content of the array using a Quick Sort algorithm. The array must be single dimensional. Returns the Sort order.

Length ( )

Returns the size of the array.

Compact ( )

Removes all of the NULL objects from the array.

Copy (Integer from, Integer count)

Copy a number of elements within the array to another array. Give the place in the index to start the copy in the from parameter and the number of elements to copy in the count parameter. An array containing these copied elements is returned.

Insert (Integer index, Object o)

Insert the element provided in `Object` o into the spot designated by the index parameter.

Remove (Integer index)

Remove the element in the array at the location of the index parameter.

8.11.1.9 Date Methods

Use `System.Date` to create a `Date` object, which contains the date and time. The following are other methods that pertain to dates:

- GetYear ( )
- GetMonth ( )
- GetDay ( )
- Format (String format)
- IsLeapYear ( )

GetYear ( )

Retrieve the year out of the `Date` object.

GetMonth ( )

Retrieve the month represented by an integer from 1 to 12 out of the `Date` object.

GetDay ( )

Retrieve the day of the week where Sunday is 0 to Saturday, which is 6, out of the `Date` object.

Format (String format)

Format the date as described by the format string, which can be either `dd/mm/yyyy` or `mm/dd/yyyy`.

IsLeapYear ( )

Check if the year of the date is a leap year or not. Returns true if it is a leap year; false if not.

8.11.1.10 Time Methods

A `Time` object represents `Time` value. A `Date` object always contains a `Time` object. You can also create a `Time` object using the `System.Time` function. In addition, the following methods pertain to time:

- GetHour ( )
- GetMinute ( )
- GetSecond ( )
- Format (String format)
- To12Hour( )

GetHour ( )

Retrieve the hour out of the `Time` object.

GetMinute ( )

Retrieve the minute out of the `Time` object.

GetSecond ( )

Retrieve the second out of the `Time` object.

Format (String format)

Format the `Time` object using the provided format string. The format string should either be `hh:mm:ss` or `hh:mm`.

To12Hour( )

Convert the `Time` object to a 12 hour format instead of a 24 hour format.

8.11.1.11 Enumeration

Contains a list of objects. Some of the object types that can be contained in the Enumeration object is a SQL result set, a String Tokenizer, or Request Parameter names.

Count ( )

Counts the number of elements in the Enumeration object. Returns the number of elements.

Next ( )

Accesses the next element in the Enumeration object.

8.11.1.12 File

An object of this type can be used to access contents of a file in the file system. You must use the `OpenFile` function to open an existing file (a `System` function). OTL does not allow creation of new files or modification of existing files.

Exists ( )

True is returned if the file exists in the file system.

Open ( )

Open the file for reading. Throws an exception if the file does not exist.

ReadLine ( )

Returns a string from the open file. It reads a line from the file that is terminated by a `\r\n`.

8.11.2 Operators That You Can Use With the Device Manager Tag Language

You can use operators for calculations on certain objects, as

Table 8–3 Device Manager Tag Language Operators

Operator	Description
+	Use can add numbers within Integer, Long, or Double objects. If applied to a String, the strings are concatenated.
-	Subtract numbers contained in Integer, Long, or Double objects. Subtract dates or time.
*	Multiply numbers contained in Integer, Long, or Double objects.
/	Divide numbers contained in Integer, Long, Double, or Character objects.
%	A mod operator applied against Integer, Long, Double, or Character objects.

8.11.3 Syntax for the Device Manager Tag Language

OTL supports all regular scripting language syntax rules, such as Assignment, Conditional Constructs, and Sub Routines.

8.11.3.1 Initialization Statements

You can define primitive data types or arrays using the following:

- Defining primitive data types: Use the `SET` syntax to define a new variable in OTL. `SET` can also be used as an Assignment statement.

  ```
  <c:set var="a" value="1"/>
  <c:set var="b" value="String variable"/>
  <c:set var="c" value="${a}"/>
  ```

- Defining an array: Use the `SET` syntax to define the array, which can hold any type of object, including mixed types.

  ```
  <c:set var="arr1" value="${{"aa", "bb", "cc"}}"/>
  ```

```
<c:set var="arr2" incex="0" value="10"/>
<c:set var="arr3" value="${{}}"/>
```

The first array, `arr1`, is initialized with the values provided. The second array, `arr2`, is initialized with the number 10 at index 0. The third array, `arr3`, creates an empty array. When values are assigned to an existing array, the array is expanded, as necessary.

The following example expands the array, `arr1`, to size 11. All of the values from index 3 to 9 is set to NULL.

```
<c:set value="${arr1.insert(10, "dd")}"/>
```

If there already was an object at location 10, then the object is replaced with the new object, "`dd`". To insert a new object at index 10 and keep existing data, use the `Insert` method, as follows:

```
<c:set value="${arr1.insert(10, "dd")}"/>
```

8.11.3.2 Assignment Statements

`SET` and `SQL` are two distinct assignment syntax statements.

- `SET` supports normal operations, such as arithmetic operations. Normal arithmetic operations can be used on most of the primitive data types, as well as other objects. OTL converts data types appropriately when arithmetic operations are applied to objects.
- `SQL` executes SQL statements on a database connection, which results in a `SQLRESULTSET` object.

```
<c:set var="a" value="1"/>
<c:set var="a" value="${a + 2}"/>
<c:set var="b" value="${"1" + 2}"/>
<c:set var="dt" value="${System.Date ("01-01-2004")}"/>
<c:set var="dt" value="${dt + 1}"/>
```

In this example, `a` is first assigned the value of 1. Then, two is added to `a`, which brings the value to three. The value of `b` is initialized to 12 and `dt` is initialized to the date of Jan. 1, 2004. Lastly, a 1 is added to `dt`, bringing the date value to 01-01-2004.

8.11.3.2.1 Creating a SQL Result Set Use the `SQL` syntax to create a SQL Result. `SQL` syntax is similar to the `SET` syntax. The following example assigns a SQL statement to the `rs` variable.

```
<c:sql var="rs" value="select table_name from all_tables"/>
```

8.11.3.2.2 Print Value to the Output Stream Use the OUT syntax to print a value to the output stream object, as follows:

```
<c:out value="${a}"/>
```

8.11.4 Conditional Statements

OTL supports the following four types of conditional statements:

- If-Else
- While
- Foreach
- Choose

Each statement must end with the appropriate end tags. Conditional operators, such as &&, ||, ==, >, >=, <, <=, and != are supported by OTL. However, implicit boolean conditions are not allowed, such as `if (value)`.

8.11.4.1 If-Else Conditional Statement

The `if-else` conditional statement enables you to execute a block of statements depending upon a condition. `ELSEIF` statements are not supported.

```
<c:if test="${a == 1 && b == 2}">
 ...
c:else/>
 ...
</c:if>
```

8.11.4.2 While Conditional Statement

The `while` statement enables you to execute a block of statements repeatedly until the condition check fails.

```
<c:while test="${a == 1}">
 ...
</c:while>
```

8.11.4.3 Foreach Conditional Statement

The `foreach` conditional statement enables to you enumerate built-in enumeration objects, such as `SQL Result Set` and `Vectors`. Also, this statement is used to execute a block of statements repeatedly by stepping through a `STEP` value.

```
<c:foreach var="row" items="${rs}">
 <c:out value="${row[0]}"/>
</c:foreach>

Then use the break statement to exit the loop:

<c:foreach var="row" items="${rs}">
 <c:if test="${row[0] begin="1" end="${count}" step="3">
   <c:out value="${row}"/>
</c:foreach>
```

8.11.4.4 Break Statement

Break from a loop with the `break` statement.

```
<c:foreach var="row" items="${rs}">
 <c:if test="${row[0] == "1"}">
   <c:break/>
</c:foreach>
```

8.11.4.5 Choose Statement

The `choose` statement supports a mutually exclusive conditional execution, where only one of a number of possible actions is executed. The following example executes one of the `when` blocks depending on `value`:

```
<c:choose>
  <c:when test="${value < 20}">
    <c:out value="Greater than 20"/>
  </c:when>

  <c:when test="${value == 20}">
```

```
    <c:out value="Equal to 20"/>
  </c:when>

  <c:otherwise>
    <c:out value="Less than 20"/>
  </c:otherwise>
</c:choose>
```

8.11.5 Define Custom Functions

You can define custom functions. These functions have a global scope from the point of definition, which means that they can access all global variables within the same OTL page. All variables defined within a function have local scope, except for the `Array` data type.

In the following example, the `par1`, `par2`, `par3`, and `local` variables have local scope. Any modifications to these variables are not relected in other parts of the script. If you want to return more than one object from a function, use the `System.SetAttribute` and `System.GetAttribute` methods.

```
<c:func var="PrintData" params="par1, par2, par3">
  <c:out value="${par1}"/>
  <c:set var="local" value="${par1}"/>
  <c:return value="${par2 + par1}"/>
</c:func>

<c:set var="a" value="${PrintData ("Function Call", 1, 2)}"/>
<c:out value="${a}"/>
```

8.11.6 Manage the Database Connection

You can use `database` to specify the database connection information used to establish a connection for the application or to disconnect from the database. Only one connection for each application is allowed in the OTL engine.

8.11.6.1 Specify Database Connection Information for an Application

Specify the database connection information used to establish a connection for the application. There is only one connection for each application

```
<c:database username="SYSTEM" password="P" DSN="POLITE""/>
```

8.11.6.2 Disconnect from the Database

At then of executing against the database, if you want to disconnect, then issue the following:

```
c:database action="disconnect"/>
```

8.11.7 Global Classes

The device manager OTL engine contains two predefined global classes, which are available to any script that access operating system and device manager information.

- Section 8.11.7.1, "Methods of the System Class": Use to access operating system information.
- Section 8.11.7.2, "Methods of the DeviceManager Class": Use to access device manager information.

8.11.7.1 Methods of the System Class

You can use the following system functions in your device manager command:

- Section 8.11.7.1.1, "Retrieve HTTP Request Parameters and Session Values"
- Section 8.11.7.1.2, "Create a Date Object"
- Section 8.11.7.1.3, "Create a Time Object"
- Section 8.11.7.1.4, "Get, Set, or Remove Session Attributes"
- Section 8.11.7.1.5, "Retrieving Parameter Name or Value"
- Section 8.11.7.1.6, "Retrieving the Request URL"
- Section 8.11.7.1.7, "Retrieving the Last Error Message"
- Section 8.11.7.1.8, "Retrieving System Memory Information"
- Section 8.11.7.1.9, "Retrieving Storage Information"
- Section 8.11.7.1.10, "URL Encoding a String"
- Section 8.11.7.1.11, "Opening a File"
- Section 8.11.7.1.12, "Synchronizing Databases"

8.11.7.1.1 Retrieve HTTP Request Parameters and Session Values You can retrieve the existing HTTP request parameters and HTTP Session values, as follows:

- Retrieve HTTP Request Parameters
- Retrieve HTTP Session Attributes

Retrieve HTTP Request Parameters

You can retrieve all of the HTTP request parameters, such as the URL and Form parameters. To retrieve a specific parameter value, prefix the parameter name with a colon—such as `:param_name`—or use the `GetParameterValue` function. All of the parameter values are preprocessed and the URLs are decoded by the tag language for you.

For example, if a URL is `c://my_app/index.html?my-Par=abcde`, then you can retrieve the parameter value in either of the following ways:

```
<c:out value="${my_par} />
```

or you can use the `GetParameterValues` function to retrieve all of the input parameters and then use `GetParameterValue` function to retrieve the value of each individual parameter, as follows:

```
<c:set var="rs" value="${System.GetParameterNames()}" />
<c:foreach var="r" items="${rs}">
 <BR>Parameter Name = <c:out value="${r} />
 Parameter Value = <c:out value="$System.GetParmaeterValue (r)}" />
</c:foreach>
```

For more information on `GetParameterValues` and `GetParameterValue`, see Section 8.11.7.1.5, "Retrieving Parameter Name or Value".

Retrieve HTTP Session Attributes

You can retrieve and store session attributes by prefixing the parameter name with a colon (`:name`) or through the Session get and set functions.

```
<c:set var="dummy" value="${System.SetAttribute ("NAME", "VALUE")}"/>
```

```
<c:set var="val1 value="${System.GetAttribute ("NAME")}"/>
<c:set var="val2 value="$(:NAME}"/>
```

As you can see, `val1` demonstrates how to retrieve the value using the `GetAttribute` function and `val2` demonstrates how to retrieve the value using `:NAME`. Substitute the actual HTTP session parameter name for `NAME`.

For more information on get and set attribute functions, see Section 8.11.7.1.4, "Get, Set, or Remove Session Attributes".

8.11.7.1.2 Create a Date Object Create a Date object with the current time using `System.Date()`. Create a Date object with a predefined date value using `System.Date(String date)`, where `date` is a date string.

8.11.7.1.3 Create a Time Object Create a Time object with the current time using `System.Time()`. Create a Time object with a predefined time value using `System.Time(String time)`, where `time` is a time string.

8.11.7.1.4 Get, Set, or Remove Session Attributes You can get, set, or remove Session attributes. To set an attribute in the application session, use `System.SetAttribute (String name, Object value)`. Each attribute has a unique name and value. To get the attribute value, use `System.GetAttribute (String name)`. To remove the attribute, use `System.RemoveAttribute (String name).`

8.11.7.1.5 Retrieving Parameter Name or Value You can retrieve all of the parameters that are provided through the `GetParameterNames` method, which returns all parameters in an `Enumeration` object. Given a parameter name, you can retrieve the value through the `System.GetParameterValue(String name)` method.

Retrieve all parameters into the `params` variable, as follows:

```
<c:set var="params" value="${System.GetParameterNames()}" />
```

Then, once you have retrieved the parameters, you can parse through them using a for loop, as follows:

```
<c:foreach var="parm" items="${params}">
 <BR>Parameter Name = <c:out value="${parm} />
 Parameter Value = <c:out value="$System.GetParameterValue (parm)}" />
</c:foreach>
```

Each parameter is read into the `parm` variable and the name is retrieved using `value="${parm}`. The value is retrieved with the `System.GetParameterValue` method.

8.11.7.1.6 Retrieving the Request URL Use the `System.GetURL()` method for retrieving the request URL.

8.11.7.1.7 Retrieving the Last Error Message Use the `System.GetError` method for retrieving the last error message. If any of the command statements resulted in an error, such as a database error while executing a SQL statement, retrieve the error using this method.

8.11.7.1.8 Retrieving System Memory Information Use `GetMemoryInfo` method to retrieve the device memory information. The following parameters are supported by this function:

- 0 - Retrieve free memory (virtual)

- 1 - Retrieve total memory (virtual)
- 2 - Retrieve free memory (physical)
- 3 - Retrieve total memory (physical)

`System.GetMemoryInfo (Integer type)`—Given a value between 0 and 3, returns a `Long` value containing the requested memory information.

8.11.7.1.9 Retrieving Storage Information Use System.GetStorageInfo to retrieve device storage information. The return value sent back is in KB.

- 0 - Retrieve free storage
- 1 - Retrieve total storage

The second parameter must be a drive name or a directory name. If the function is invoked without parameters, then the function retrieves the free storage space in the root directory.

`System.GetStorageInfo(Integer type, String drive)`—returns a `Long` value containing storage information.

8.11.7.1.10 URL Encoding a String Encodes the provided string.

System.URLEncode (String value)

Returns the given string as a URL encoded string.

8.11.7.1.11 Opening a File Use `System.OpenFile (String name)` to open the file provided in the method parameter. Returns the `File` object.

8.11.7.1.12 Synchronizing Databases Use the `System.CreateSyncClient` method to create a `Synchronization` client object. Call the `Synchronization.Synchronize` method to synchronize. Retrieve any error messages using the `System.GetError` method.

```
<c:set var="sync " value="${System.CreateSyncClient()}"/>
<c:set value="${sync.SetUserName ("S11U1")}"/>
<c:set value="${sync.SetPassword ("manager")}"/>
<c:set value="${sync.SetServerURL ("http://localhost")}"/>
<c:set value="${sync.SetProxyInfo ("www-proxy:80")}"/>
<c:set var='ret' value="${sync.Synchronize()}"/>
<c:if test="${ret != 0}">
   <BR>Synchronization error = <c:out value="${System.GetError()}"/>
</c:if>
```

8.11.7.2 Methods of the DeviceManager Class

The device manager methods are accessed using `DeviceManager.FunctionName` syntax. These functions can only be used by trusted OTL scripts.

- DeviceManager.UploadFile (File file, String URL)
- DeviceManager.GetServerURL ( )
- DeviceManager.GetBinaryDir ( )
- DeviceManager.GetUserName ( )
- DeviceManager.CreateRequest (String cmd)
- DeviceManager.GetRegistry (String key, String name)
- DeviceManager.SetRegistry (String key, String name, String value)

- DeviceManager.LogMessage (String handler, String name, String message)

DeviceManager.UploadFile (File file, String URL)

Use `UploadFile` method to upload a file to the Mobile Server, which contains the device management server. In order to use this method, you must first successfully use `System.OpenFile` on the file in question.

Given a `File` object and a URL, returns true if the upload is successful.

DeviceManager.GetServerURL ( )

Returns the URL of the Mobile Server.

```
<c:set var='url' value='${DeviceManager.GetServerURL()}'/>
```

DeviceManager.GetBinaryDir ( )

Returns the full path of the binary directory of Oracle Database Lite client.

```
<c:set var='dir' value='${DeviceManager.GetBinaryDir()}'/>
```

DeviceManager.GetUserName ( )

Returns the Oracle Lite username

```
<c:set var='user' value='${DeviceManager.GetUserName()}'/>
```

DeviceManager.CreateRequest (String cmd)

Create a Device Management request (or command) and notify Device Management Agent to process it. Command string must have corresponding OTL script file in the client device.

The following example demonstrates notifying the DM Agent to process and OTL script of the name: `sync.otl`.

```
<c:set value='${DeviceManager.CreateCommand ("sync")}'/>
```

DeviceManager.GetRegistry (String key, String name)

Retrieve a value from the Oracle Lite configuration file (`POLITE.INI`). All the values are retrieved as `String`. The following example retrieves the value for `DATA_DIRECTORY` configured in the `POLITE.INI` file.

```
<c:set var='val' value='${DeviceManager.GetRegistry ("All Databases", "DATA_
DIRECTORY")}'/>
```

DeviceManager.SetRegistry (String key, String name, String value)

Set a new configuration value in the Oracle Lite configuration file (`POLITE.INI`). The following example sets a new value for `DATA_DIRECTORY`.

```
<c:set value='${DeviceManager.SetRegistry ("All Databases", "DATA_DIRECTORY",
"C:\TEMP")}'/>
```

DeviceManager.LogMessage (String handler, String name, String message)

Log a message in the Device Manager logging system. The Device Management client uploads all logged messages to the Oracle Lite Mobile Server.

```
<c:set value='${DeviceManager.LogMessage (0, "My Log", "Log message..."}'/>
```

Applications may use this method to send data to the server. In order to accomplish this, you must create a handler on the server to process the client message. Once created, you must register this handler in the Mobile Server. If your application is written in C/C++, JAVA or Visual Basic, you may use corresponding native APIs to log any message. For a C/C++ application, use the following:

```
dmLogMessage (const TCHAR* handler, const TCHAR* name, const TCHAR* message);
```

In order to use the above API, you must dynamically load the `OMCAPI.DLL` library and extract the function pointers. The following sample code demonstrates how to log a message:

```
typedef void (*dm_Initialize)();
typedef void (*dm_Destroy)();
typedef void (*dm_LogMessage) (LPCTSTR, LPCTSTR, LPCTSTR);
HMODULE hMod = ::LoadLibrary (TEXT ("omcapi.dll"));
if (hMod)
{
  dm_Initialize init =
         (dm_Initialize)::GetProcAddress (hMod, TEXT ("dmInitialize"));
  dm_Destroy    dest = (dm_Destroy)::GetProcAddress (hMod, TEXT ("dmDestroy"));
  dm_LogMessage log  =
         (dm_LogMessage)::GetProcAddress (hMod, TEXT ("dmLogMessage"));
  if (init && dest && log)
  {
    (*init)();
    (*log) (TEXT ("MY_HANDLER"), TEXT ("MY LOG"), TEXT ("My Message"));
    (*dest) ();
   }
   ::FreeLibrary (hMod);
}
```

If you want to use the Java API, then set `WEBTOGO.JAR`, which is part of 'Oracle Lite WEB' client. The following code sample demonstrates how to log a message with Java APIs:

```
import oracle.lite.dm.ClientAPI;
public class MyLog
{
  public static void main (String[] args)
  {
    ClientAPI.initialize();
    ClientAPI.logMessage ("MY_HANDLER", "MY LOG", "My Message");
    ClientAPI.destroy();
  }
}
```

On the server-side, a message handler must be developed and registered. A message handler is a Java class that implements the `oracle.lite.provider.MessageListener` interface. See the Javadoc for more information on the `MessageListener` interface. The following is an example of a message handler:

```
import oracle.lite.resource.Device;
import oracle.lite.provider.MessageListener;
import oracle.lite.provider.MessageData;
public class MyMessage implements MessageListener
{
  public void initialize (String metaData) throws Exception {}
  public void destroy() throws Exception {}
  public void service (Device device, String name, MessageData data)
           throws Exception
  {
    // Process 'My Message'
    }
}
```

Once the message handler is implemented and compiled, copy the JAR file to *<ORACLE_HOME>*`\Mobile\class` directory. Then, execute the following SQL script to register your message handler implementation:

```
msql mobileadmin/manager@jdbc:oracle:thin:@<mobile_server_db>:<port>:<sid>
insert into dm$all_providers values ('MY_HANDLER', 'MESSAGE', 'MyMessage', NULL);
```

8.11.8 Importing Another OTL Page

Use the `import` statement to include a page into the current page. All URL parameters are available to scripts in the imported page, as well as in the current page below the point of inclusion.

```
<c:import url="url_of_the_include_page"/>
```

Specify URL parameters using the HTTP format with the `?` or the `<c:param>` tag, as shown below:

```
<c:import url="URL?abc=def">
  <c:param name="name1" value="value1"/>
  <c:param name="name2"> value2 </c:param>
</c:import>
```

8.11.9 Installing an Application From a Server

You can install a new application in an Oracle mobile client by downloading all of the files specified in the `download` statement, as shown below, and then register the application with the mobile client.

```
<c:install name="Sample App" url="/sample1/home.htm" icon="/sample1/icon.gif"
        server="http://server_name:8081/download" enable='Y' version="1.0.0.2">
   <c:download name="home.htm" url="app/home.htm" value="10202020"/>
</c:install>
```

8.11.10 Specifying a Schema File

If you cannot establish a connection to the database or find the table designated when executing a `sql` statement, then OTL finds a `schema` statement to locate the OTL script that creates the table and establishes the database connection. The following `schema` statement establishes that the OTL file is `/app1/schema.otl`.

```
<c:schema url="/app1/schema.otl"/>
```

If a table is missing in the database while processing a `<c:sql>` statement, then the OTL engine retrieves the schema file and processes it. The schema file should have script to create the missing table, such as the `create table` statement.

8.11.11 Error Handling

You can throw and catch exceptions within any OTL script. The only restriction is that you can only have at most one catch block. The throw script terminates the current script processing and jumps to the catch block. If a page contains a single throw tag and no catch tag, then the script stops upon reaching the throw tag.

```
<c:throw value='1'/>
...
<c:catch var='ex'/>
```

```
...
</c:catch/>
```

8.11.12 Sample Device Manager Commands Using the Tag Language

Retrieve current date and time:

```
<c:set var="d" value="${System.Date()}"/>
<c:out value="Date = ${d}"/>
<br><c:out value="Time = ${d.Time()}"/>
```

Perform Date arithmetic by first retrieving the date, then adding or subtracting 2 days from it. The last two lines changes the date to either be 2 days from now or 2 days ago.

```
<c:set var="d"  value="${System.Date()}"/>
<c:set var="d2" value="${d + 2}"/>
<c:set var="d3" value="${d - 2}"/>
Date + 2 = <c:out value="${d2}"/>
<br>Date - 2 = <c:out value="${d3}"/>
```

Formatting date and time by retrieving the time using the `Date` or `Time` methods, and then applying a format. For the date, apply either day/month/year or month/day/year with the `Format` method. For time, you can choose the format of hours:minutes.

```
<c:set var="d"  value="${System.Date()}"/>
Date (dd/mm/yyyy) = <c:out value="${d.Format ("dd/mm/yyyy")}"/>
Date (mm/dd/yyyy) = <c:out value="${d.Format ("mm/dd/yyyy")}"/>
<c:set var="t" value="${System.Time()}"/>
Time (hh:mm) = <c:out value="${t.Format ('hh:mm')}"/>
```

You can apply the names of the month or day to a date by using `GetMonth` and `GetDay`, as follows:

```
<c:set var="month" value="${{"Jan", "Feb", "Mar", "Apr", "May", "Jun",
            "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"}}"/>
<c:set var="day" value="${{"Sunday", "Monday", "Tuesday", "Wednessday",
            "Thursday", "Friday", "Saturday"}}"/>
<c:out value="${month[d.GetMonth() - 1]}"/>
<c:out value="${day[d.GetDay()]}"/>
```

Retrieve the day and add two days to it:

```
<c:set var="d" value="${d + 2}"/>
```

Set the date to 02-20-2004.

```
<c:set var="d" value="${date ("02-20-2002")}"/>
```

9 Offline Instantiation

The Offline Instantiation (OLI) utility enables Mobile administrators to prepare a package that 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.

The following sections discuss the Offline Instantiation feature.

9.1 Using Offline Instantiation to Distribute Multiple Mobile clients

You can enable your users to install their client using a distribution method, such as a CD, through the network, or email. To install the Mobile client and perform the first synchronization to retrieve the applications (with the initial data) can be a performance issue. In this case, the administrator pre-creates the Mobile binaries with the user ODB files (includes the applications and data for the user) to the client. The download of this package is faster than having each user perform the first synchronization on their device. Thus, this procedure helps users avoid an expensive performance hit when creating and synchronizing the Mobile client for the first time.

Offline instantiation is a tool that enables an administrator to gather and package the Mobile client binaries and the user applications and data into a single directory. Offline instantiation is part of the Mobile Development Kit, which can be installed only on a Windows platform. Thus, you create all of your user distribution files on a Windows machine and you can only create multiple user distribution files for Web-to-Go, Branch Office, Win32, and WinCE Mobile clients. We recommend that you use the same Windows environment where a Mobile server exists to create your distribution files.

When you have multiple users who use the same application, you set up a distribution for each user through the following steps:

1. Using the Mobile Manager on the Mobile Server, the administrator sets up the application and the users for the Mobile client distribution, as follows:

    a. Using the Mobile Manager on the Mobile Server, the administrator publishes the applications that are to be installed on the Mobile clients.

    b. The administrator creates all of the users for the Mobile clients.

    c. The administrator grants access for these users to the applications that are to be downloaded for the distribution.

2. On the Windows machine where the Mobile Development Kit is installed, perform a normal client installation and synchronization for one of your users as a template for the client distribution. Choose the platform and language that are appropriate for the Mobile clients that you are creating, except for the WinCE platform. To set up the Mobile client on the WinCE (PocketPC) device, choose the Oracle Lite WIN32. For steps on creating a Mobile client, see Section 2.1, "Preparing the Device for a Mobile Application" and Section 2.2, "Installing the Mobile Client Software".

3. Download the Mobile client binaries—without the application or user data—in a unique directory. See Section 9.3, "Downloading the Client Binaries Into the Client Root Directory" for directions.

4. Configure the `oli.ini` file to tell the offline instantiation batch tool how to create the client distribution packages. See Section 9.5, "Configure How OLI Creates the Client Distribution Packages With the OLI Configuration File" for directions.

5. Use the offline instantiation tool (OLI) to create and package the final client distribution packages for each user. See Section 9.6, "Using the OLI Engine to Create and Package the Client Distribution File" for directions.

6. Distribute the client distribution packages to each user to install on their client device. See Section 9.7, "Deploying Client Distribution Files on Client Machines" for directions.

9.2 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, are 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`

9.3 Downloading the Client Binaries Into the Client Root Directory

To download the Mobile client binaries into the Mobile client root directory, the administrator performs the following:

1. Download `setup.exe` from the Mobile Server setup page, which is available at the following URL:

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

   Choose the appropriate Oracle Database Lite Mobile client. For example, choose Oracle Lite Branch Office for Branch Office applications, Oracle Lite WEB for Web applications, Oracle Lite WEB BC4J for Web applications that use BC4J, and the Oracle Lite WIN32 for WinCE (Pocket PC) and Win32 applications.

   You use the same `setup.exe` that installs a client, but instead, install the client into a directory that will be used as a template for all of the clients you will be distributing.

2. Run the `setup.exe` using the distribution (`-d`) flag as follows.

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

   > **Note:** This program creates a `setup.exe` icon in the same folder to which you downloaded the executable. Running this without the `-d` flag, by double-clicking the icon, for instance, performs a normal setup of the client on your machine. The `-d` option must be used to create the necessary files for an offline install.

3. Enter your user name and password.

4. The **Choose Directory** screen prompts you for a destination directory where you want to create the client bundles. Ensure that you create a new destination directory other than the directory in which the `setup.exe` file resides. The appropriate files are copied to the destination directory.

   The install directory specified during this process is defined as `MOBILECLIENT_CD_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.

   > **Note:** Ensure that the `MOBILECLIENT_CD_ROOT` distribution directory is different from the `MOBILECLIENT_ROOT` directory.
   >
   > See Section 9.5, "Configure How OLI Creates the Client Distribution Packages With the OLI Configuration File" for information on the elements in the `oli.ini` file.

5. Finally, download the cilent binaries from the Mobile Server by successfully synchronizing the Mobile client 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.

Note: This only downloads the binaries for the Mobile client. The user data and applications are installed with the OLI process, as defined in Section 9.7, "Deploying Client Distribution Files on Client Machines".

9.4 The OLI Package Directory

The OLI package directory is a location where all the individual 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`.

See Section 9.5, "Configure How OLI Creates the Client Distribution Packages With the OLI Configuration File" for information on the elements in the `oli.ini` file.

Note: Ensure that this directory is different from `MOBILECLIENT_CD_ROOT` and `MOBILECLIENT_ROOT`.

9.5 Configure How OLI Creates the Client Distribution Packages With the OLI Configuration File

The offline instantiation tool, OLI, reads the `oli.ini` file to determine how many users, the names of those users, the location of the Mobile client binaries, and so on. Before you use the offline instantiation tool, make sure that you set up these parameters correctly.

The offline instantiation tool and configuration file is located in the Mobile Development Kit, under *`ORACLE_HOME`*`\Mobile\Sdk\bin\`. Thus, make sure that you have installed the Mobile Development Kit.

The following describes how to configure the `OLI.INI` file:

- Section 9.5.1, "SETUP"
- Section 9.5.2, "USERS"
- Section 9.5.3, "Example of OLI.INI File"

9.5.1 SETUP

The `SETUP` section contains the general configuration for OLI to create the client packages.

MOBILE_SERVER

Provide the Mobile Server host and port that the Mobile clients will connect to for synchronization. You can supply a host name or an IP address. The default port number is 80. Sync server must be running when OLI is launched.

```
MOBILE_SERVER=myhost.us.oracle.com:80
```

USE_SSL

If you are going to use SSL, set to YES. Default is NO.

```
USE_SSL=NO
```

JDBC_URL

Configure the JDBC URL to the Mobile Server Repository.

```
JDBC_URL=jdbc:oracle:thin:@myhost.us.oracle.com:1521:oid
```

SCHEMA and PASSWORD

Configure the Mobile Server administration schema name and password for the Mobile Server Repository.

```
SCHEMA=MOBILEADMIN_SCHEMA
PASSWORD=MOBILEADMIN_PASSWORD
```

MAKEODB_METHOD

The MAKEODB_METHOD parameter defines how the client Oracle Lite databases are populated. The default and more performant option is JDBC, which transfers the data for all clients from the repository over a JDBC connection. Otherwise, you can configure SYNC, which uses the client-server synchronization for each individual client to generate the Oracle Lite databases.

```
MAKEODB_METHOD=JDBC
```

OLITE_JDBC_DRIVER

The OLITE_JDBC_DRIVER parameter defines the JDBC driver type for the connections to Olite client databases. Valid values are NATIVE and ODBC.

- NATIVE: The default; use the Oracle Database Lite native driver.
- ODBC: use the SUN JDBC-ODBC bridge.

```
OLITE_JDBC_DRIVER=NATIVE
```

MOBILECLIENT_ROOT

OLI needs to know where you installed the template Mobile client. This is the client that you installed in Section 9.3, "Downloading the Client Binaries Into the Client Root Directory". This Mobile client is for a user that is identical to all of the other users that will use this distribution. Thus, OLI uses this Mobile client as the template for how to build the distribution.

```
MOBILECLIENT_ROOT=C:\MOBILECLIENT
```

MOBILECLIENT_CD_ROOT

OLI needs to know where you installed the Mobile client binaries when you executed `setup -d`. Configure the directory that you specified in Section 9.3, "Downloading the Client Binaries Into the Client Root Directory". OLI packages these binaries with the same configuration as designated by the Mobile client installed in the MOBILECLIENT_ROOT directory for each user defined in the USERS section.

```
MOBILECLIENT_CD_ROOT=C:\MOBILECLIENT_CD
```

SHARED_CD_MODE

If set to YES, then only one generic client CD is generated and placed in the <OLI_CDS_ROOT>/Shared_CD. This CD only contains shared data. If set to NO, then each user has its own package created under the <OLI_CDS_ROOT>/<username>.

OLI_CDS_ROOT

This is the final directory where OLI places all of the client distribution packages for each user. From this directory, you can distribute the packages to each user.

```
OLI_CDS_ROOT=C:\OLI_CDS
```

DEVICE_TYPE

This specifies the type of device to which the client distribution packages are installed. You cannot install the packages to multiple platforms. Instead, choose one of the following:

- WIN32: Windows 32
- WTG: Web-to-Go client, Branch Office
- WCE: Windows CE (PocketPC)

```
DEVICE_TYPE=WIN32
```

THREADS

You can specify the number of threads OLI can use to process all of the users listed in the OLI.INI file. The more threads you allow, the more users can be processed concurrently.

9.5.2 USERS

The USERS section defines the users and their passwords. For each user, OLI creates a client distribution package that contains the Mobile client binaries as well as the user applications and data. The clients must have been created on the Mobile Server, as described in Section 9.1, "Using Offline Instantiation to Distribute Multiple Mobile clients". On each line, put the username and password, as follows:

```
[USERS]

CONSC1 MANAGER
CONSC2 MANAGER
```

For each user, a client distribution package is created.

9.5.3 Example of OLI.INI File

The following sample configuration file is available on the MDK host at *ORACLE_HOME*\Mobile\Sdk\bin\.

```
########################################################################
#
# 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 package.
# This is the destination directory when you run "setup.exe" with an option "-d".
#
MOBILECLIENT_CD_ROOT=C:\MOBILECLIENT_CD

# If set to YES only one generic client CD is generated and place in
# <OLI_CDS_ROOT>/Shared_CD. This CD only contains shared data. If set to NO,
# each user has its own package created under <OLI_CDS_ROOT>/<USERNAME>.
#
SHARED_CD_MODE=NO
```

```
#
# 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
#
[USERS]
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
```

9.6 Using the OLI Engine to Create and Package the Client Distribution File

The OLI engine reads the file `oli.ini` in the current directory for information related to configuration settings. Before launching the OLI engine, you must edit the `oli.ini` file, as described in Section 9.5, "Configure How OLI Creates the Client Distribution Packages With the OLI Configuration File". The OLI engine provides 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.

Note: Before launching the OLI engine, you must do the following:

- Ensure that the JDK version is 1.4 or later.
- Ensure that the mobile devices for each user that will be a part of the OLI are NOT disabled.
- 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.

  See Section 9.5, "Configure How OLI Creates the Client Distribution Packages With the OLI Configuration File" for information on the elements in the oli.ini file.

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 Java class used by OLI, 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.

Note: Shut down the Web-to-Go client prior to executing the oli.bat file.

This action displays the following usage information. NOTE: You execute only ONE of the following commands at a time: makeodb, package, cleanup, or check status. Do NOT execute oli.bat with more than one of these commands. You will notice that the instructions show how to create the offline instantiation packages by executing oli.bat several times—once for each command.

```
Usage
-----
oli.bat [-d] [-i] [-g] [makeodb] [package] [cleanup] [check status]
-g  Debug mode
```

The command options for oli.bat are as follows:

- -d: If your oli.ini configuration file (see Section 9.5, "Configure How OLI Creates the Client Distribution Packages With the OLI Configuration File") is not in the default directory, specify where it is located with this flag.
- -i: If you renamed the oli.ini file to another name, specify the name with this option.
- -g: Turn on debugging.

To carry out OLI tasks, re-execute the command using the appropriate switches and arguments.

To build the client installation package, perform the following:

1. Section 9.6.1, "Create and Populate Client Database Files with the MAKEODB Command"

2. Section 9.6.2, "Package the Mobile Client Binaries with the Client Database Files with the package Command"
3. Section 9.6.3, "Clean Up the OLI Tables Before Executing OLI for Another Distribution"
4. Section 9.6.4, "Check the Status of OLI Clients"

> **Note:** Before executing the makeodb and package commands on WinCE or Win32 devices, ensure that you set the DEVICE_TYPE parameter to WCE, WTG, or Win32 in the oli.ini file.

9.6.1 Create and Populate Client Database Files with the MAKEODB Command

Creates and populates the client Oracle Lite database 'odb' files. For each user defined in the [USERS] section of the oli.ini file, OLI creates the client database files for subscribed publications.

Usage

```
oli.bat [-g] makeodb
```

The initial status of a client is RESET. After the first client is processed successfully by makeodb, then its status changes from RESET to SLUG. After all of the other clients are replicated using the first client, their status changes from RESET to ODBMADE. Finally, when the OLI engine packages the client information into a directory, the status changes to PACKAGE.

You can see the state of the client distribution files by executing the checkstatus command (see Section 9.6.4, "Check the Status of OLI Clients").

9.6.2 Package the Mobile Client Binaries with the Client Database Files with the package Command

After creating the client database file, execute the package command, which packages up the client database file and the Mobile client binaries for each user defined in the [USERS] section in the oli.ini file. Each client package is written to a subdirectory of the client name under the directory defined in the OLI_CDS_ROOT section in the oli.ini file.

Usage

```
oli.bat [-g] package
```

After a client package is successfully processed, its status is changed to PACKAGED. You can see the state of the client distribution files by executing the checkstatus command (see Section 9.6.4, "Check the Status of OLI Clients").

9.6.3 Clean Up the OLI Tables Before Executing OLI for Another Distribution

The cleanup command cleans the OLI tables. The cleanup command re-creates the OLI tables. Execute this command before executing OLI for another distribution. Do not execute this command without a need to start again.

Usage

```
oli.bat [-g] cleanup
```

9.6.4 Check the Status of OLI Clients

Check the status of OLI clients with the checkstatus command.

Usage

```
oli.bat checkstatus
```

9.7 Deploying Client Distribution Files on Client Machines

Once you have the client packages ready, you can distribute them to your users either by putting the distribution files on a CD for them or by giving the user access to the distribution files over the network or through email. Whether you use the CD or provide your users access to the distribution directory, the client must have network access to the Mobile Server. When using OLI to register the client, the connection is used to propagate the initial synchronization of data.

When you finish packaging the users using the OLI command, a directory is created in the OLI_CDS_ROOT directory for each user. In each subdirectory, the distribution files, with a setup.exe, is written. The user can execute the setup.exe directly from this subdirectory over a network, you can zip up all of these files and send the ZIP file to the user over email, or you can copy all of the files to a CD for each user. Once the user has access to the distribution files, the user executes the setup.exe to install the Mobile client and the user applications and data.

The deployment process for WinCE applications are different from those of native Win32 applications and Web-to-Go applications. The following sections describe how the user would install the distribution files on these devices:

- Section 9.7.1, "Deploy Win32 Native or Web-to-Go Client Distribution Package"
- Section 9.7.2, "Deploy WinCE PocketPC Client Distribution Package"

9.7.1 Deploy Win32 Native or Web-to-Go Client Distribution Package

To deploy on client devices for native Win32 applications or 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 sub-directory (package) is named after the client name. Provide each user with the client distribution package; for example, copy the client package to the client machine.
2. On the client device, perform the following:
   a. If you have a Mobile client installed, uninstall the existing software.
   b. From the client distribution package, run setup.exe.

9.7.2 Deploy WinCE PocketPC Client Distribution Package

To deploy on PocketPC client devices 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 sub-directory (package) is named after the

client name. Provide each user with the client distribution package. Copy the client package to the *ORACLE_HOME* directory of the Pocket PC device.

3. Perform a synchronization.

9.8 Creating a Single Package for Users That Share Data

Instead of creating a separate package for each user, with its own data, you can create a single package for users that share data. Then, once installed, the user can log on and retrieve any data that is unique to the user.

Follow the same directions as if you were creating the multiple packages, with the following differences:

1. In the Mobile Server configuration, configure the MAGIC_CHECK parameter in the Data Synchronization Instance parameters. See Section 6.3, "Configuring Data Synchronization" on where the Instance parameters are modified in the Mobile Manager. Modify the MAGIC_CHECK parameter to be NONSHARED. See the description of the MAGIC_CHECK parameter in Section B.6, "[CONSOLIDATOR]".
2. Configure only a single username in the list of users at the end of the oli.ini file. See Section 9.5, "Configure How OLI Creates the Client Distribution Packages With the OLI Configuration File" for an example.
3. Configure the SHARED_CD_MODE element in the oli.ini file to YES. See Section 9.5, "Configure How OLI Creates the Client Distribution Packages With the OLI Configuration File" for an example.
4. After you complete the packaging of the offline instantiation by executing the oli.bat package command (see Section 9.6, "Using the OLI Engine to Create and Package the Client Distribution File"), then copy the contents of the <OLI_CDS_ROOT>/SHARED_CD directory onto a CD.
5. Place the CD into the CD drive of the client device, change directory to the CD and execute the following:

   Note: The client must have network access to the Mobile Server. When using OLI to register the client, the connection is used to propagate the initial synchronization of data.

   ```
   setup.exe setup.inf <username> <password>
   ```

6. After you have finished installing all of your clients and have performed the first synchronization for each of them, change the MAGIC_CHECK parameter in the Data Synchronization Instance parameter bakc to the default of ALL.

10 Manage Your Branch Office

The following sections describe how to install, configure, manage and use the Mobile client for Branch Offices:

- Section 10.1, "Introduction"
- Section 10.2, "Branch Office Installation and Configuration"
- Section 10.3, "Architecture"
- Section 10.4, "Administration"

> **Note:** If you have installed Branch Office 10g Release 1 and want to use version 10g Release 2, you must perform some upgrade steps that are listed in the Upgrade chapter of the *Oracle Database Lite Getting Started Guide*.

10.1 Introduction

The following sections introduce the Oracle Database Lite Branch Office:

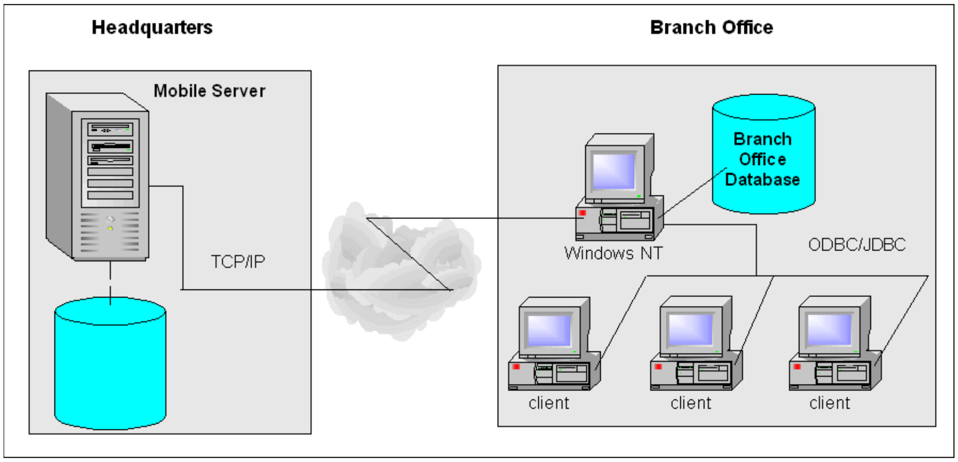
- Section 10.1.1, "What is the Branch Office?"
- Section 10.1.2, "How the Branch Office Works"
- Section 10.1.3, "The Branch Office Manager"
- Section 10.1.4, "Synchronizing Data with Headquarters"

10.1.1 What is the Branch Office?

The Branch Office provides access to the Branch Office database for up to 32 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 10–1 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 headquarters.

Figure 10–1 Branch Office Overview



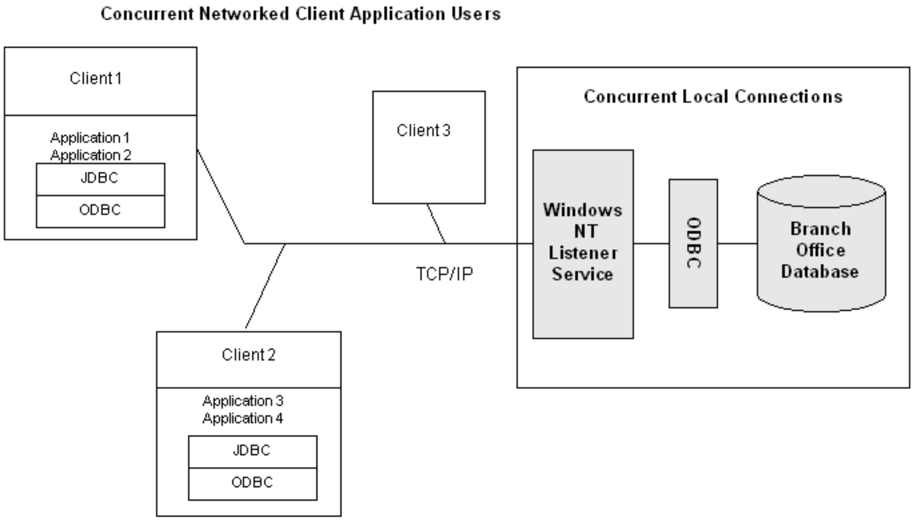
10.1.2 How the Branch Office Works

Each Branch Office database supports up to 32 concurrent networked users, which are also known as 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 32 concurrent local ODBC/JDBC connections to the Branch Office database. These local connections can be used for applications that perform background tasks, such as reporting, mass changes or updates, and bulk data loading.

As Figure 10–2 displays, Branch Office clients and local ODBC and JDBC applications can access the Branch Office database simultaneously. Multiple applications can execute on each client.

Figure 10–2 Accessing the Branch Office Database



10.1.3 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 the Web-based Branch Office Mobile Manager interface, the Branch Office Administrator centrally manages Branch Office operations. Furthermore, by using the Branch Office Mobile Manager, the administrator does not need to be physically present at each Branch Office.

10.1.4 Synchronizing Data with Headquarters

Data synchronization for Branch Office is centrally managed by the Branch Office Administrator. The Administrator synchronizes applications and data with the database located at 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 Chapter 6, "Managing Synchronization" and the Synchronization chapter in 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.

10.2 Branch Office Installation and Configuration

The following sections describe how to install and configure the Mobile client for Branch Office:

- Section 10.2.1, "Terms and Concepts"
- Section 10.2.2, "Overview"
- Section 10.2.3, "Branch Office Installation"
- Section 10.2.4, "Enabling Branch Office on Windows XP Service Pack 2"
- Section 10.2.5, "Changing Branch Office Listener Port Number"

10.2.1 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 maching that contains the Branch Office.

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

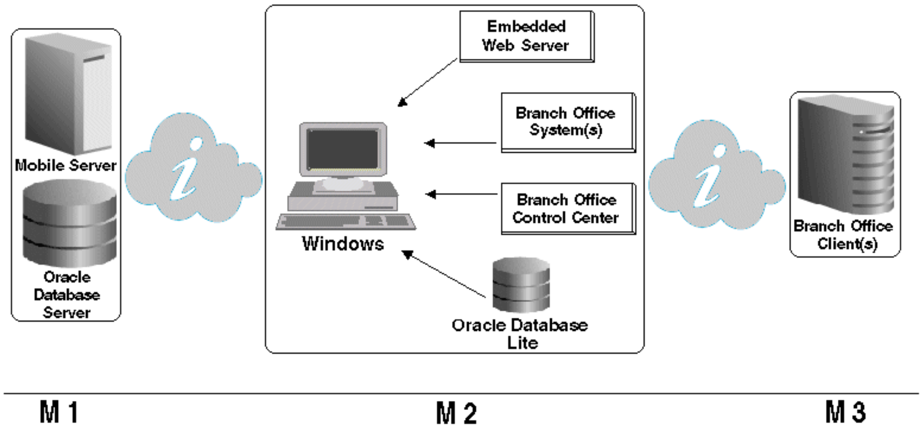
Group of Branch Office Administrators, each managed by the Mobile Manager.

10.2.2 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 10–3 displays, this example 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: Up to the maximum of thirty-two Branch Office clients that host the Branch Office application (.exe) and use a remote ODBC/JDBC connection to access data in the Branch Office (multi-user) database located on M 2.

Figure 10–3 Branch Office Setup



10.2.3 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 to the Packaging Wizard chapter in the *Oracle Database Lite Developer's Guide*.

3. Using the Mobile Manager Applications page, publish your Branch Office application. Select the Branch Office application that you need to publish and click **Publish Application**.
4. Using the Mobile Manager Users page, create a Branch Office Administrator user and add this user to the *Branch Administrators* group. Provide administrator privileges to the Branch Office Administrator user.
5. Using the Mobile Manager Applications page, click the published Branch Office application link. Select the Files tab and choose the application files that you want installed on the Branch Office Client (M3) machine. Click **Make Public** to specify that these files are public files.
6. Provide access privileges to the Branch Office Administrator user. Using the Mobile Manager Applications page, select the Branch Office application that you need to provide user access privileges for and click **Access**. 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>
   ```

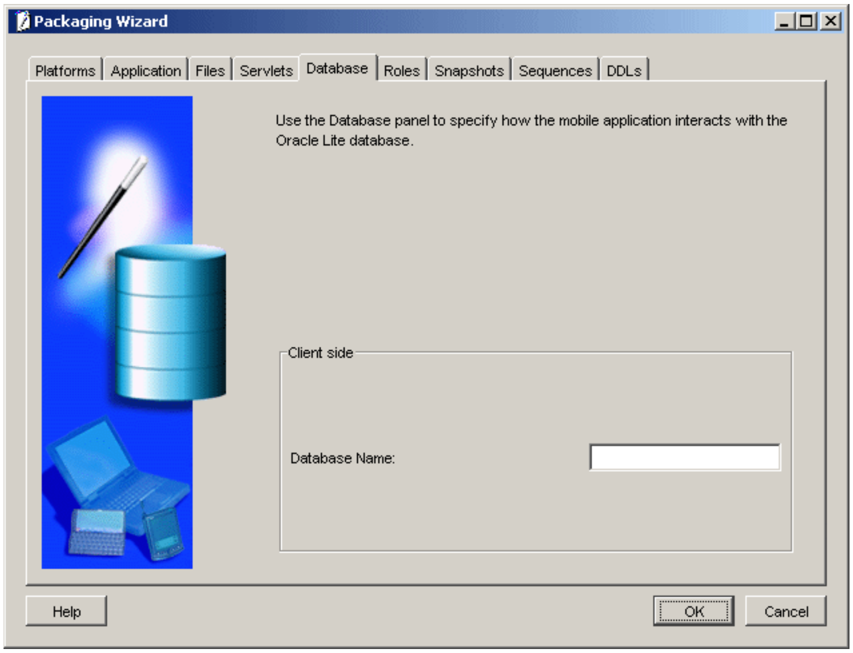
   OR

   ```
   http://localhost
   ```

   > **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 `oldb40` directory 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 10–4.

Figure 10–4 Database Panel - Packaging Wizard



Use the Database panel to specify how the mobile application interacts with the Oracle Lite database.

Client side

Database Name:

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`.

10. Open the Branch Office Mobile Manager and create a new Branch Office user.

    This stage marks the conclusion of the Branch Office Installation and Configuration process. In the next step, you must configure the Branch Office Client (M3).

11. 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).

12. 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
    ```

13. On the Branch Office machine M3, add `oladc12040.dll` and `olc12040.dll` to your path.

14. If the client created in step 11 is boUser, then when you connect from the client to the Branch Office, the remote JDBC connection string for the user named boUser is shown below.

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

    The DSN name is the same as the name provided in step 9 in the <user name>_<database name> format 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 located on the Branch Office machine (M2). The default port number for the database listener can be modified by modifying the port number in the SERVICE_PORT parameter in the POLITE.INI file.

10.2.4 Enabling Branch Office on Windows XP Service Pack 2

When you install Windows XP Service Pack 2, the Internet Connection Firewall (ICF) defaults to ON. In order for the Branch Office Server to work properly, you either need to turn the ICF OFF or enable port 1531 within the ICF. To enable port 1531, go to the Windows Firewall control on your Windows machine. Select the Exception tab. Click **Add Port**. Add port 1531 with any name.

10.2.5 Changing Branch Office Listener Port Number

In order to change the Branch Office Server Listener port number, perform the following steps:

1. Stop both the '**Oracle Web-to-go**' and the '**Oracle Lite Multiuser**' services.
2. In the polite.ini file, edit the [All Databases] section to include the SERVICE_PORT parameter, which points to the new listening port.

   For example:

   ```
   [All Databases]
   SERVICE_PORT=1531
   ```

3. Start the '**Oracle Lite Multiuser**' service first and then the '**Oracle Web-to-go**' service.

> **Note:** The sequence in which services are started and stopped should be in the order as described above.

10.3 Architecture

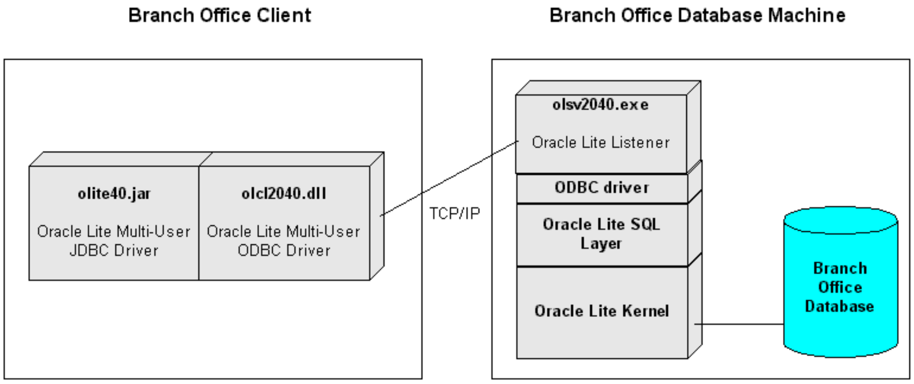
The following sections describe the components of the multi-user architecture:

- Section 10.3.1, "The Branch Office Environment"
- Section 10.3.2, "Connecting Clients to the Branch Office Database Machine"

10.3.1 The Branch Office Environment

The Branch Office environment is comprised of two parts. As Figure 10–5 displays, they are the Branch Office Client component and the Branch Office Database component.

Figure 10–5 The Branch Office Environment



10.3.1.1 The Branch Office Client

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

> **Note:** The `branch.odb` file represents a sample database used throughout this chapter. Your database name may be different.

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 (*`OLITE_HOME`*`/bin/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.

10.3.1.2 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`.

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

Note: Snapshots are owned by `SYSTEM`. The password is the Branch Office administrator 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). 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 platform executes 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 that 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.

10.3.1.3 Company Headquarters

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

10.3.2 Connecting Clients to the Branch Office Database Machine

The client applications connect with the Branch Office database machine through 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.odb`.

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

10.3.2.1 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 10–1 describes the above database connection statement.

Table 10–1 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/<username>` folder in the *`ORACLE_HOME`* directory.

10.3.2.2 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 10–2 describes the above Branch Office database connection statement.

Table 10–2 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.lite.poljdbc.POLJDBCDriver");
                  conn = DriverManager.getConnection
                 ("jdbc:Polite@DATA_SERVER:1531:POLITECL","SYSTEM","MANAGER");
}
   catch(Exection 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.

10.4 Administration

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 and users. Navigate to the Mobile Workspace to administer the Branch Office applications and replication jobs. The Branch Office Administrator is a Mobile Server user created by the Mobile Server Administrator and must be included as a member of the group, "BRANCH ADMINISTRATORS."

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.

The following sections provide instructions for using the Branch Office Mobile Manager:

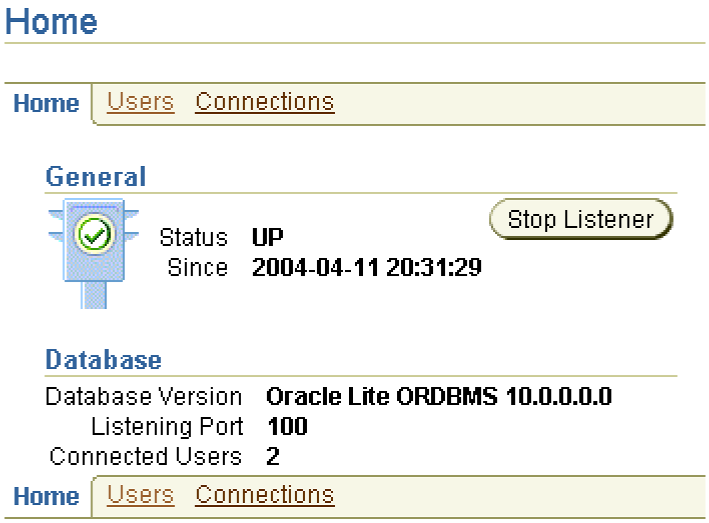
- Section 10.4.1, "Logging into the Branch Office Manager"
- Section 10.4.2, "Using the Branch Office Manager"
- Section 10.4.3, "Managing Branch Office Users"
- Section 10.4.4, "Managing Applications"

10.4.1 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 10–6.

Figure 10–6 Branch Office Home Page



Status UP

Since 2004-04-11 20:31:29

Database Version Oracle Lite ORDBMS 10.0.0.0.0

Listening Port 100

Connected Users 2

10.4.2 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.

10.4.2.1 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.

10.4.2.2 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.

10.4.2.3 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.

10.4.2.4 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 10–3 describes the Branch Office home page.

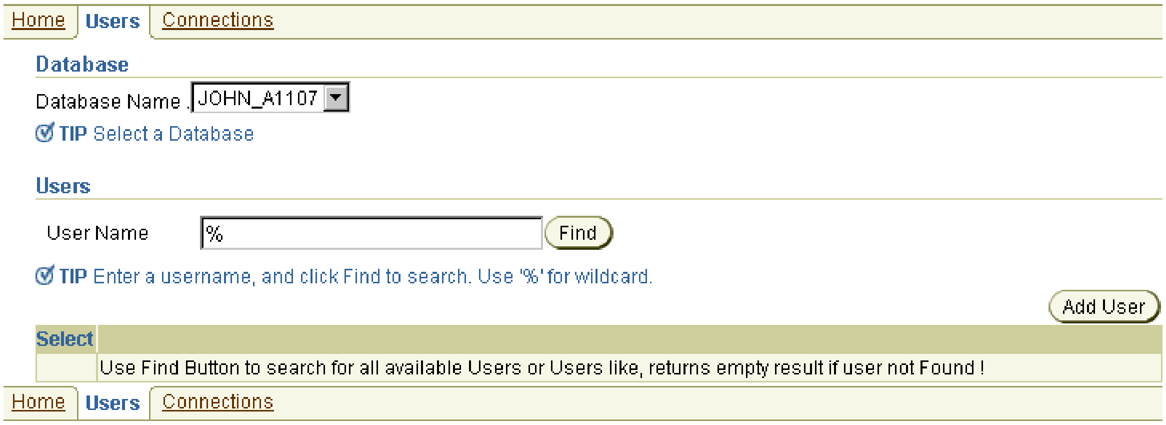
Table 10–3 Branch Office Home Page Description

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.

10.4.3 Managing Branch Office Users

To manage Branch Office users, login to the Mobile Server and navigate to the Users page. As Figure 10–7 displays, the Users page appears.

Figure 10–7 Branch Office Users Page



Database

Database Name JOHN_A1107

TIP Select a Database

Users

User Name %

TIP Enter a username, and click Find to search. Use '%' for wildcard.

Select	
	Use Find Button to search for all available Users or Users like, returns empty result if user not Found !

The Branch Office database does not need to be stopped to manage users and their access privileges. The Branch Office Administrator can add or delete users while other users are accessing the Branch Office database.

10.4.3.1 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.

10.4.3.2 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 10–4 describes, the Branch Office Administrator can assign the following roles.

Table 10–4 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.

10.4.3.3 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.

10.4.3.4 Setting User Privileges

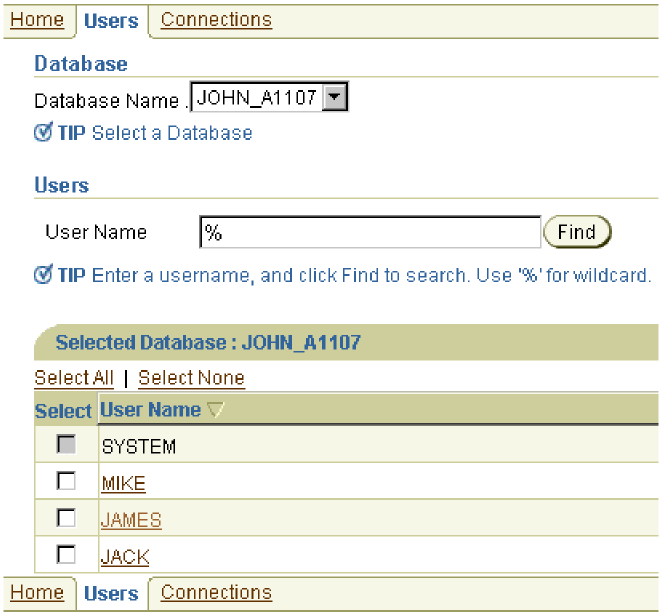
The Privileges page enables the Branch Office Administrator to assign user privileges. To control user access to database tables, you can grant user privileges such as Select, Delete, Insert, and Update.

10.4.3.5 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 10–8 displays, the Users page displays users that are associated with the chosen database.

Figure 10–8 Displaying Branch Office Users



Database

Database Name JOHN_A1107

TIP Select a Database

Users

User Name %

TIP Enter a username, and click Find to search. Use '%' for wildcard.

Selected Database : JOHN_A1107

Select	User Name
☐	SYSTEM
☐	MIKE
☐	JAMES
☐	JACK

10.4.3.6 Removing a User

To remove a user, select the check box displayed against a user name and click the Delete button.

10.4.4 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.

- Section 10.4.3.1, "Creating Users"
- Section 10.4.3.2, "Setting User Roles"
- Section 10.4.3.3, "Setting User Properties"
- Section 10.4.3.4, "Setting User Privileges"
- Section 10.4.3.5, "Finding Users"
- Section 10.4.3.6, "Removing a User"

10.4.4.1 Downloading Public Files to Your Client

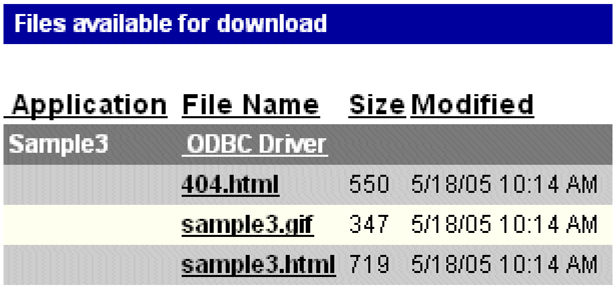
Making application files public enables the Branch Office to download those files and install them on the Branch Office clients.

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, as Figure 10–9 shows. 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.

Figure 10–9 Listing of Public Files



Files available for download

Application	File Name	Size	Modified
Sample3	ODBC Driver		
	404.html	550	5/18/05 10:14 AM
	sample3.gif	347	5/18/05 10:14 AM
	sample3.html	719	5/18/05 10:14 AM

For ODBC configuration, click the ODBC driver link. This downloads the `setup.exe`. After the file is downloaded, users must run the `setup.exe`.

11

Using the Application Server OID With Mobile Server

If you decide to use OID in OracleAS, then after you install the application server and Oracle Mobile Lite, perform the following:

1. Migrate any existing Mobile users to OID (See Section 3.6, "Migrate Your Users From the Mobile Server Repository to the Oracle Internet Directory" in the *Oracle Database Lite Getting Started Guide*).
2. Use the Mobile Manager to edit the configuration file (`webtogo.ora`) and set `SSO_ENABLED=YES`. Do not edit the `webtogo.ora` file directly.
3. Restart the application server.
4. Navigate to the Users tab in the Mobile Manager, which displays all users in OID. Select the users to enable for Mobile Server and enable these users. Assign the appropriate application to these users.
5. Install the Mobile client and synchronize using one of the users you enabled.

12

Manage Security in Oracle Database Lite

The following sections detail how to manage security in Oracle Database Lite:

- Section 12.1, "Configuring SSL For Mobile Server"
- Section 12.2, "Encrypting the Client Oracle Lite Database"
- Section 12.3, "Security Warning for Demo Applications"

Note: There is additional information about developing for security in the "Security" chapter in the *Oracle Database Lite Developer's Guide*.

12.1 Configuring SSL For Mobile Server

Oracle Database Lite 10*g* 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.

- Section 12.1.1, "Creating an SSL Certificate"
- Section 12.1.2, "Configuring Mobile Server for SSL"
- Section 12.1.3, "Troubleshooting Error Messages for an SSL-Enabled Mobile Server"
- Section 12.1.4, "Client-Side Configuration for Secure Socket Layer (SSL)"

Note: These are server-level steps which are typically executed prior to deployment of an application that requires SSL communication.

12.1.1 Creating an SSL Certificate

Oracle Database Lite ships a sample keystore file with a self-signed sample certificate. The password for this sample keystore file is `oracle`. 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/tooldo` 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/
```

12.1.2 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

12.1.2.1 Configuring SSL for Mobile Server With Oracle9*i*AS or OracleAS

For production systems, you have either Oracle9*i*AS 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 Oracle9*i*AS 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.

12.1.2.2 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 `http-web-site.xml` to *`ORACLE_HOME`*`j2ee\mobileserver\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 *`ORACLE_HOME`*`j2ee\mobileserver\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 `http-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 `http-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
   ```

12.1.3 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 `http-web-site.xml` files.

12.1.4 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.

The following sections describe how to enable SSL for your Mobile client:

- Section 12.1.4.1, "Communication between the Mobile Client and the Mobile Server"
- Section 12.1.4.2, "Connection between the Browser and the Mobile Client for Web-to-Go"
- Section 12.1.4.3, "Support for Non-SSL Mobile Clients"

12.1.4.1 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`.

12.1.4.2 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 Oracle9*i*AS.
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`

12.1.4.3 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 the Oracle Application Server, the Mobile Server runs on both SSL and non-SSL ports, to support SSL and non-SSL clients. The application server must be configured to run on both SSL and non-SSL ports, as a default function.

12.2 Encrypting the Client Oracle Lite Database

Whether you are using a Mobile client or an embedded application, you can encrypt the database used on the client.

- When encrypting the Oracle Lite database used with the Mobile client, modify the `POLITE.INI` file to set the `ENCRYPT_DB` parameter. See Section H.3.2.13, "ENCRYPT_DB" for more information.
- When encrypting the Oracle Lite database that is used with an embedded application, use the `encryptdb` executable, as described in the "ENCRYPDB" section Appendix B in *Oracle Database Lite Developer's Guide*.

12.3 Security Warning for Demo Applications

If you have the demo applications installed in a production environment, they can be used to access areas of Oracle Database Lite that you may want to be secure. The demo applications are provided for you to use when learning how to develop your own application. Thus, when you are finished developing your product, remove the demo applications from the repository. For directions, see the Installation chapter in the *Oracle Database Lite Getting Started Guide*.

13 Performance

Tuning options for your applications are described in the following sections:

13.1 Using Connection Pooling for Applications

Connection pooling enables you to eliminate the time delay in creating and destroying connections for incoming application requests. Instead, enable connection pooling, as shown in Section 4.3, "Manage Application Properties or Users", so that each incoming connection request uses an existing connection from the pool.

13.2 Limit Application Connection Requests to the Database

You can limit the number of connections that access the database from each application, as shown in Section 4.3, "Manage Application Properties or Users". Set the maximum database connection limit. Any request for a database connection beyond the limit is refused.

13.3 Determine Correct Synchronization Templates to Use for Subscriptions

You can use the Consperf performance tool to evaluate if the default templates used with your publication items are the best performance option. See Section 6.7, "Monitoring and Analyzing Performance" for a full description.

13.4 Synchronization Tablespace Layout

Tablespace layout across multiple disks can improve the performance of Mobile Server data synchronization, as it reduces movement of the disk heads and improves I/O response time.

By default, the synchronization tablespace is `SYNCSERVER`, and is stored in the `mobilexx.dbf` file in the default location for the database instance under *`ORACLE_HOME`*, where *`xx`* is a number between 1 and 25. The tablespace name, filename, and file location for the tablespace is defined in the `$OLITE_HOME/Mobile/Server/admin/consolidator_o8a.sql` script file, which is executed during the Mobile Server installation process. So, if you want to modify the tablespace name, filename or file location, perform the following BEFORE you install the Mobile Server; otherwise, the default tablespace is created.

- To modify the name of the tablespace, locate every instance of `SYNCSERVER` and change it to the name of your choice. For example, if you wanted the tablespace name to be `mysynctbl`, then you would modify `consolidator_o8a.sql`, as follows:

  ```
  execute immediate 'create tablespace mysynctbl datafile '''||filename||''' size
  200m autoextend on';
  ```

- To modify the name of the datafile where the tablespace is stored, locate the following line in `consolidator_o8a.sql`:

  ```
  filename := 'mobile'||filename||'.dbf';
  ```

  This is included within a loop that assigns a number between 1 and 25 as an extension to `mobile`, so that the resulting filename is between `mobile01.dbf` to `mobile25.dbf`. You can modify the name of the file by modifying `mobile` to the name of your choice. For example, if you wanted the name to be `syncdata01.dbf`, then do the following:

  ```
  filename := 'syncdata'||filename||'.dbf';
  ```

- To designate an absolute or relative directory of where the tablespace datafile is located, modify the execution statement in `consolidator_o8a.sql` to specify the absolute or relative file location. The default places the datafile in the database instance directory in *`ORACLE_HOME`*. For example, if you are on a Windows platform and you want the datafile to be located on the `e:\syncserver` directory, modify the execute statement, as follows:

  ```
  execute immediate 'create tablespace syncserver datafile
  ''e:\syncserver\'||filename||''' size 200m autoextend on';
  ```

  This places the datafile into the `E:\syncserver` directory. If you wanted the file to be in a directory relative to the default location, provide a relative pathname. For example, to have the datafile located in a directory below the database instance in its own directory called `synctbl`, modify the execute statement, as follows:

  ```
  execute immediate 'create tablespace syncserver datafile
  ''./sycntbl/'||filename||''' size 200m autoextend on';
  ```

13.5 Synchronization Performance Affected by WebCache

If you decide to use WebCache with Oracle Database Lite, the performance of any synchronization you initiate from a Web-to-Go client is significantly slower—approximately anywhere from 30 seconds to 2.5 minutes.

14

Configure for National Language Support (NLS)

In order to support a language that contains multi-byte characters, perform the following:

- Section 14.1, "Configuring OC4J to Handle Multibyte Characters in Online Web Applications"

14.1 Configuring OC4J to Handle Multibyte Characters in Online Web Applications

If you have an application that uses multibyte characters and runs in online mode, you need to configure the `default-charset` element to the machine locale in the OC4J `global-web-application.xml` file to allow multibyte characters. For example, a Japanese machine should have its locale set to the `Shift_JIS` locale in the OC4J `global-web-application.xml` file to allow Japanese multibyte characters, as follows:

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

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

15 Backup and Recovery

Performing backup and recovery for Oracle Database Lite is the same as what you would normally do for Oracle database applications. The following sections help you understand how to use the Oracle database backup and recovery methods for preserving your Mobile Server and Mobile applications:

15.1 How Does Oracle Database Lite Store its Information?

Oracle Database Lite uses the Oracle database to store information, as follows:

- The Mobile Server itself is installed and configured as a database application. Thus, the Mobile Server stores its metadata and client state information within a database schema.
- For each Mobile application, the Mobile Server installs triggers and stores transaction data in a schema for that application.

15.2 Backing Up Oracle Database Lite

Since all of the data needed for a backup and recovery strategy exists in the database, you should use the Oracle database backup and recovery strategies discussed in the following books:

- *Oracle Backup Installation Guide*
- *Oracle Database Recovery Manager Quick Start Guide*
- *Oracle Database Backup and Recovery Basics*
- *Oracle Backup Administrator's Guide*
- *Oracle Database Backup and Recovery Advanced User's Guide*

> **Note:** In the past, we recommended that you use export/import to perform a backup. This is not a recommended option anymore. Use the normal online Oracle database backup procedure.

However, the following sections describe what to keep in mind when coming up with a backup and recovery strategy for your Oracle Database Lite environment: Section 15.3, "Oracle Database Lite Backup Coordination Between Client and Server" and Section 15.4, "Oracle Database Lite Recovery Issues".

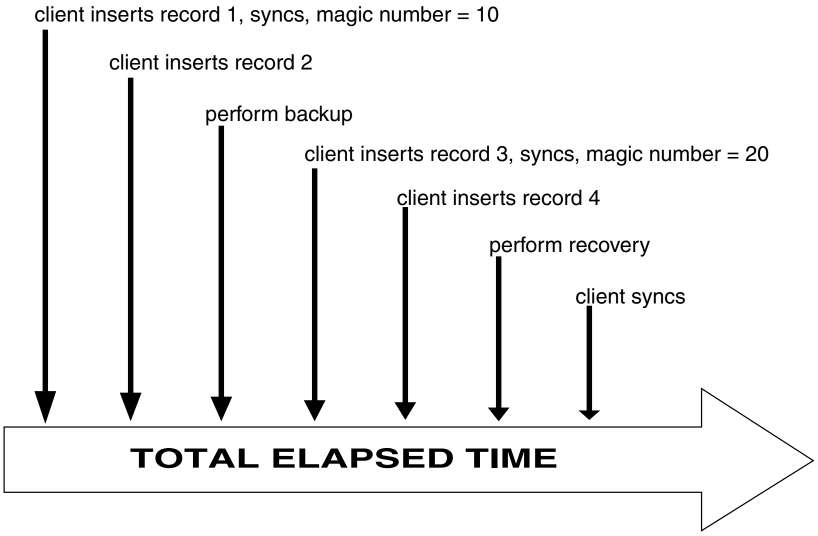
15.3 Oracle Database Lite Backup Coordination Between Client and Server

When a client and a server synchronize with each other, the Mobile Server assigns the same "magic" number to both sides to indicate that the data is in-sync. If this number is different on both sides, the Mobile Server knows that the data is out of sync and is in an error condition. The following example details how this could effect your attempts for a clean recovery.

When you perform a backup, you may lose client data unless you plan accordingly. Figure 15–1 demonstrates the following scenario:

1. The client inserts record 1 and synchronizes the data to the server. The Mobile Server assigns the same magic number to both the client and the server to denote that they are in sync. In this example, the magic number on both the client and the server is 10.
2. The client inserts record 2. No synchronization is performed.
3. The backup is performed. Client record 1 is saved to the backup. The latest magic number on both the client and the server is 10.
4. The client inserts record 3 and synchronizes the data to the server. The Mobile Server assigns the same magic number to both the client and the server to denote that they are in sync. In this example, the magic number on both the client and the server is 20.
5. The client inserts record 4. No synchronization is performed.
6. A failure occurs and the last backup is used to recover the Mobile Server, the Mobile applications and the application data. In this scenario, only record 1 is in the backup, so it will exist in the restored database.
7. The client synchronizes. Records 2 and 3 can be lost, because they are not in the backup. The msync client does not send them to the server, since they were already sent in step 4. However, the msync client does send record 4 to the server, since it is a new record that has never been synchronized with the database. After the synchronization, record 4 is stored in the error queue, not in the application tables.

Figure 15–1 Lost Data With Backup and Recovery Strategy



The Mobile Server checks the magic numbers on both the client and the server. It verifies the state of the data on the client to determine what action to take. When the client performs the next synchronization, if the magic numbers are not the same, then the following occurs:

1. The client checks if there are any new records—whether newly inserted, modified, or deleted—on the client. If so, then these records are sent to the server, which saves these records in the error queue.
2. A full refresh of all of the subscribed data is sent to the client.

In our example, if you did nothing, the client would send record 4 to the server, which would end up in the error queue, and records 2 and 3 would be lost. To save records 2 and 3, do the following:.

1. On the server, retrieve and restore the last backup.
2. On the Mobile client that is out of sync, update any record that has been modified since the last synchronization. In our example, you would do any sort of update that makes the record seem to contain new information in records 2 and 3. For example, you could update the `VARCHAR` field with the same content.
3. Initiate a synchronization on the client. The Oracle Database Lite software detects that the client database is out of sync and that some of the records have been modified. Thus, the following occurs:
   a. The modified records are updated in the restored database on the server and saved in the error queue.
   b. The server pushes a full refresh down to the client.
4. In order for you to reapply the modified records to the applications table, you must first modify the DML operation from Error to Update. The DBA must modify the record in the error queue for the base table, named `CEQ$<base_table_name>`, changing the DML operation from Error (`E`) to Update (`U`) or Insert (`I`).
5. Once updated to Update, re-execute the command. Navigate to the Error Queue screen in the Mobile Manager. Click on the modified record. Click Execute.

Note: For more information on the error queue and how to reapply the records, see Section 3.8.3 Resolving Conflicts Using the Error Queue in the *Oracle Database Lite Developer's Guide*.

6. The next time that the MGP runs, the update is applied to the application table.

Thus, all information contained within records 2 and 3 will be restored from the device.

15.4 Oracle Database Lite Recovery Issues

When you perform a recovery, the state of both the Mobile Server and the Mobile application schemas must be in-sync. If they are out-of-sync, severe problems may occur. Therefore, when you perform a backup and restoration for the Mobile Server and the Mobile application schemas, each must be recovered to the same point in time. Use the Oracle database Point-in-Time Recovery strategy to ensure that both the Mobile Server schema and Mobile application schemas are recovered to the same point in time.

The Mobile application schemas usually reside on the same Oracle database instance as the Mobile Server. However, if you have used a database link to store the Mobile application schemas on a separate Oracle database instance, then you must use a backup and restore strategy for distributed database systems.

16 Tracing and Logging

You can enable tracing for either the Mobile Server or for your Mobile Client. In addition, you can view the log files from the underlying application server. The methods for enabling tracing in each is described in the following sections:

- Section 16.1, "Enable Tracing on the Mobile Server"
- Section 16.2, "Enable Tracing on Mobile Clients"
- Section 16.3, "Viewing the Log Files From the Application Server"

16.1 Enable Tracing on the Mobile Server

For the Mobile Server, there are two main sections for tracing: the general tracing for Mobile Server components and specific tracing for data synchronization components. How to enable tracing for each part of the Mobile Server is described in the following sections:

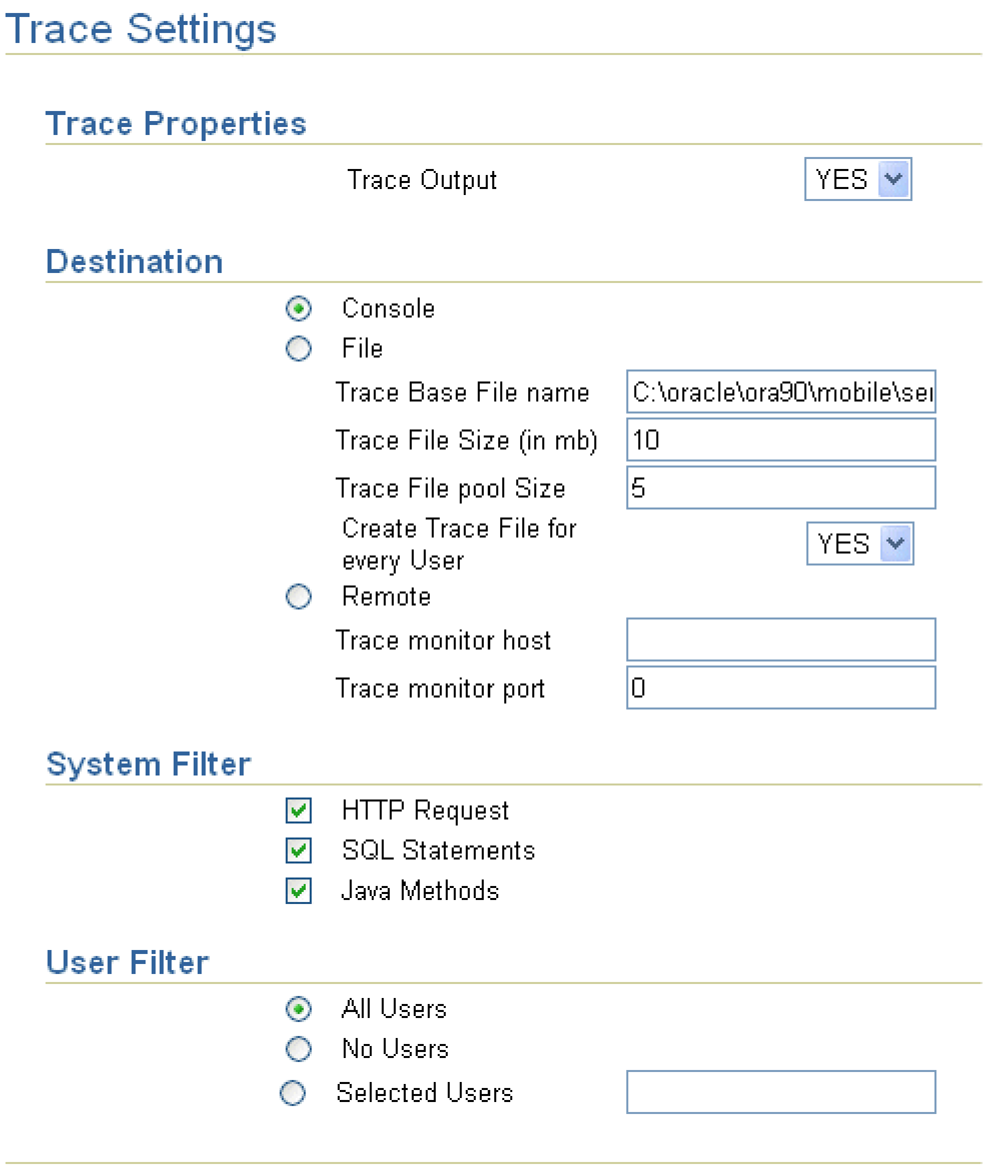
- Section 16.1.1, "General Tracing for the Mobile Server"
- Section 16.1.2, "Data Synchronization Tracing"

16.1.1 General Tracing for the Mobile Server

To set general tracing for the Mobile Server, perform the following steps.

1. From the Mobile Server page, select **Administration**.
2. Select **Trace Setting**. This brings up the Trace Settings page, as shown in Figure 16–1, where you can choose to generate trace output, specify the trace output destination to the local console, file, or remote console (viewed by `wsh`). The Trace Settings page provides system filters to generate trace output to the required system level.
3. Configure the type of tracing you want and click **Apply**.

Figure 16–1 General Trace Settings for Mobile Server



Trace Settings

Trace Properties

Trace Output: YES

Destination

● Console

○ File

Trace Base File name: C:\oracle\ora90\mobile\se

Trace File Size (in mb): 10

Trace File pool Size: 5

Create Trace File for every User: YES

○ Remote

Trace monitor host:

Trace monitor port: 0

System Filter

☑ HTTP Request

☑ SQL Statements

☑ Java Methods

User Filter

● All Users

○ No Users

○ Selected Users

Table 16–1 Trace Settings Page Description

Field	**Description**
Trace Output	To generate trace output, select **Yes**.
Console	You can print the messages to a console. You can ONLY choose the console if you are executing Mobile Server in standalone mode. If you are in an Oracle9*i*AS or OracleAS environment, select File or Remote.

Table 16–1 (Cont.) Trace Settings Page Description

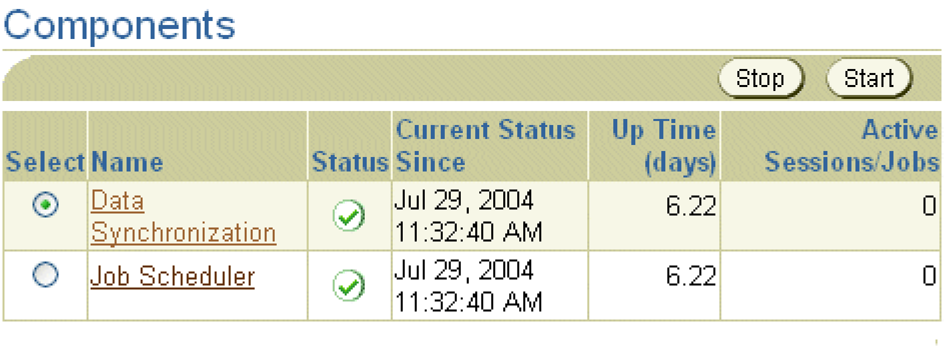
Field	Description
File	You can direct all messages to a local file. If you selected a file for trace output, then enter the name (including path), the maximum size of the file in MB, and the number of files allowed (pool size). For example, if you set the pool size to 10, then when a trace file hits the maximum size in MB, then a new file is opened and the trace output is written to the new file. This continues until all 10 files of the maximum size exist. At this point, the first file is deleted and a new file is started to contain the trace output. This enables you to manage the amount of disk space that the trace files can use. To create a trace file for every user, select **Yes** for the Create Trace File for Every User box.
System Filter	■ HTTP Request—To generate HTTP output and Web-to-Go trace information as trace output, select this option. This includes general system information. ■ SQL Statements—To generate SQL queries as trace output, select this option. ■ Java Methods—To generate all `system.out` output from the Mobile Server and Data Synchronization Java methods, select this option. **Note**: The Mobile Server automatically filters exceptions and errors as trace output at the Mandatory level.

16.1.2 Data Synchronization Tracing

The administrator can turn on tracing for components involved in the synchronization phase, including MGP functions.

1. From either the home page or the Administration page for the Mobile Server, select **Data Synchronization** in the Components section, as shown in Figure 16–2.

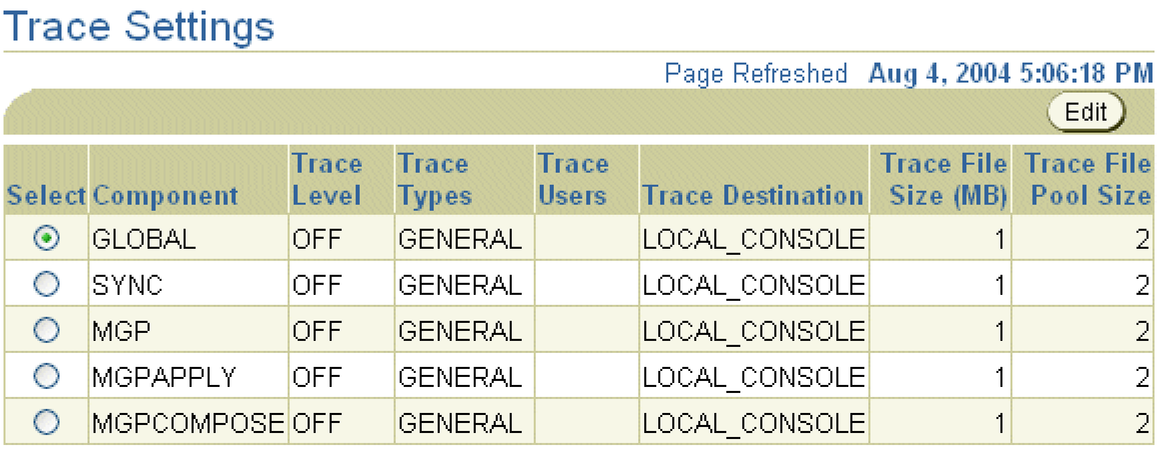
Figure 16–2 Mobile Server Job Scheduler and Data Synchronization Components



Select	Name	Status	Current Status Since	Up Time (days)	Active Sessions/Jobs
●	Data Synchronization	✓	Jul 29, 2004 11:32:40 AM	6.22	0
○	Job Scheduler	✓	Jul 29, 2004 11:32:40 AM	6.22	0

2. Select **Administration** off of the Data Synchronization home page.
3. Select **Trace Settings**, which displays all five components for which you can enable tracing, as shown in Figure 16–3. For a description of each component, see Section 16.1.2.1, "Description of the Five Data Synchronization Components".

Figure 16–3 The Trace Components for the Data Synchronization



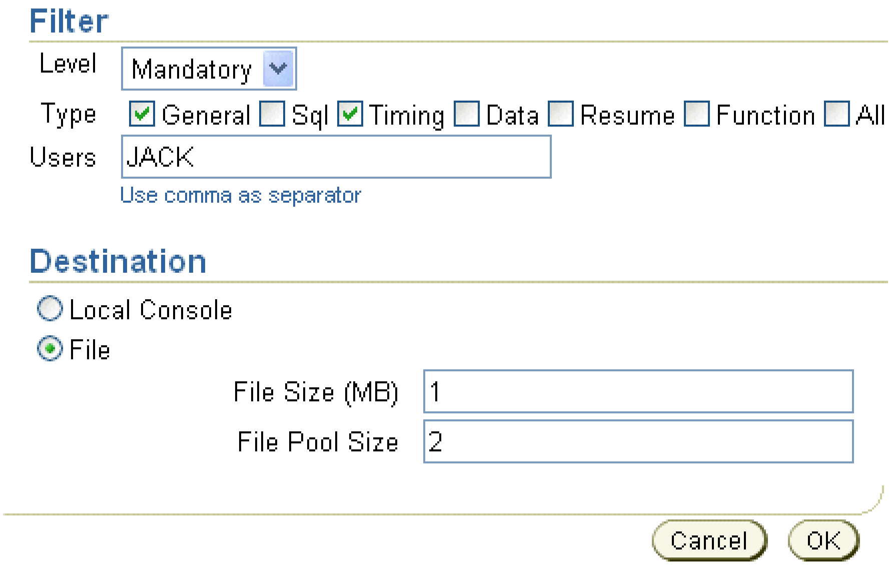
Trace Settings

Page Refreshed Aug 4, 2004 5:06:18 PM

Select	Component	Trace Level	Trace Types	Trace Users	Trace Destination	Trace File Size (MB)	Trace File Pool Size
●	GLOBAL	OFF	GENERAL		LOCAL_CONSOLE	1	2
○	SYNC	OFF	GENERAL		LOCAL_CONSOLE	1	2
○	MGP	OFF	GENERAL		LOCAL_CONSOLE	1	2
○	MGPAPPLY	OFF	GENERAL		LOCAL_CONSOLE	1	2
○	MGPCOMPOSE	OFF	GENERAL		LOCAL_CONSOLE	1	2

4. Select the component for which you want to enable tracing, which brings up the trace configuration screen, as shown in Figure 16–4.

Figure 16–4 Data Synchronization Component Trace Configuration



Filter

Level: Mandatory

Type: ☑ General □ Sql ☑ Timing □ Data □ Resume □ Function □ All

Users: JACK

Use comma as separator

Destination

○ Local Console

● File

File Size (MB): 1

File Pool Size: 2

- 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.

Table 16–2 Data Synchronization Component Trace Level and Type

Filter	Description
trace level, where each level includes the previous levels as well.	OFF: no tracing enabled. MANDATORY: Mandatory messages only, such as program exceptions. WARNING: Warning messages. NORMAL: Normal messages of which the user must be informed. INFO: Informational messages, such as synchronization timing, MGP apply, MGP compose, and MGP status. CONFIG: Configuration messages, such as JDBC driver version. FINEST: Developer level of tracing. ALL: Logs messages for all trace levels.
trace type	SQL: SQL-related messages only, such as SQL statements. TIMING: Timing data only. **Note**: This option is trace level sensitive. For MGP Cycle time and Synchronization time, use the Trace Level INFO option with the TIMING option on the MGP and SYNC components respectively. DATA: Data only. RESUME: Logs messages with Reliable Transport. FUNCTION: Displays the program flow by logging methods such as Entry, Exit or Invoke. For Long methods, this option logs the method entry or exit; which is a simple invoke log. GENERAL: Logs messages that do not belong to any of the above listed trace types. **Note**: This type is trace level sensitive. ALL: This option generates logs of all trace types.

Note: See Section B.6, "[CONSOLIDATOR]" for more details on these settings.

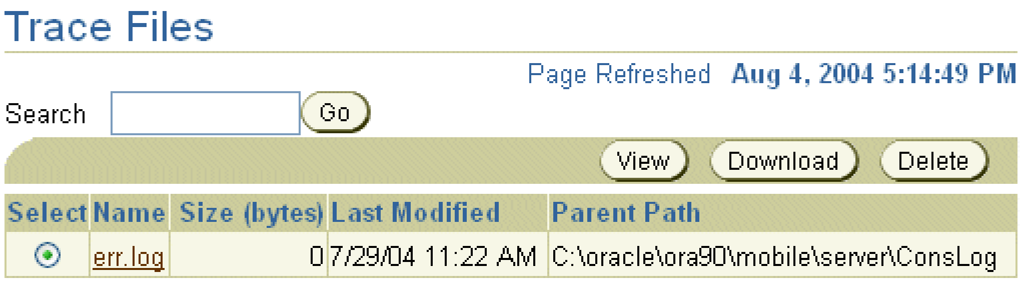
- In the Destination section, select **Local Console** to receive the trace file to the same console as the General tracing is using. If the console is not open, then these messages are sent to the same place that the General tracing is directed. See what the Destination is configured to in Figure 16–1 to determine where these messages are directed.

  To send trace information to a file, select the **File** option. The file name is generated based upon the session id. You can configure the file size in MB and the files allowed (pool number). For example, if you set the pool size to 10, then when a trace file hits the maximum size in MB, then a new file is opened and the trace output is written to the new file. This continues until all 10 files of the maximum size exist. At this point, the first file is deleted and a new file is started to contain the trace output. This enables you to manage the amount of disk space that the trace files can use.

5. To implement the modified values, click **OK**. To retain existing values, click **Cancel**.

To view trace files, navigate to the Data Synchronization page. Select **Administration**. Select **Trace Files** and the Trace Files screen appears, as shown in Figure 16–5.

Figure 16–5 Viewing Data Synchronization Trace Files



Trace Files

Page Refreshed Aug 4, 2004 5:14:49 PM

Search

Select	Name	Size (bytes)	Last Modified	Parent Path
●	err.log	0	7/29/04 11:22 AM	C:\oracle\ora90\mobile\server\ConsLog

- To view a trace file, select the trace file name or click the Select button next to the trace file name and click **View**.

> **Note:** When you view the trace file online, it truncates the file to 10,000 lines. To view the whole trace file, download the file and view it using any text editor.

- To download or delete a trace file, click the Select button next to the trace file name and click either **Download** or **Delete**.
- If there are too many files to view on a page, you can search by entering the name of the trace file in the Search field and clicking **Go**.

16.1.2.1 Description of the Five Data Synchronization Components

There are five components that you can turn on to describe what is happening in the synchronization process, as described in the following sections:

- Section 16.1.2.1.1, "MGP"
- Section 16.1.2.1.2, "MGPAPPLY"
- Section 16.1.2.1.3, "MGPCOMPOSE"
- Section 16.1.2.1.4, "SYNC"
- Section 16.1.2.1.5, "GLOBAL"

16.1.2.1.1 MGP You can trace the MGP process. However, if an `MGP Cycle ID` is not yet available, then tracing is enabled by the configuration of the `GLOBAL` component. If the trace destination is to be written to a file, then all of the generated logs are recorded in a log file named `MGP_<cycle_id>.log`.

16.1.2.1.2 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, tracing is enabled by the configuration of the component MGP. If tracing is sent to a file, then all messages are written to a file named `MGPAPPLY_<client_id>_<cycle_id>.log`.

16.1.2.1.3 MGPCOMPOSE This refers to the `COMPOSE` phase in the MGP process. Similar to the `MGPAPPLY` phase where the Client ID is not yet available, tracing is enabled by the configuration of the component MGP. If tracing is sent to a file, then all messages are written to a file named `MGPCOMPOSE_<client_id>_<cycle_id>.log`.

16.1.2.1.4 SYNC This refers to the server-side synchronization process. When a `Sync session ID` is not yet available, tracing is enabled by the configuration of the

GLOBAL component. If the trace destination is set to file, then the messages are written to a file named SYNC_<cycle_id>.log. When the Client ID becomes available, the file is renamed to SYNC_<client_id>_<cycle_id>.log.

16.1.2.1.5 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 the ConsolidatorManager APIs. If the trace destination is set to file, then the messages are written to a file named GLOBAL_<file_number>.log.

16.2 Enable Tracing on Mobile Clients

You can also enable tracing on your Mobile client through one of the following methods:

- Section 16.2.1, "Turn on Tracing using the Mobile Client WEBTOGO.ORA File"
- Section 16.2.2, "Turn on Tracing using the -d0 Option for Web-to-Go Clients With the WEBTOGO Executable"

16.2.1 Turn on Tracing using the Mobile Client WEBTOGO.ORA File

You can enable tracing through the DEBUG section in the webtogo.ora file on your Mobile Client. This is only valid for Mobile Client for the Web (Web-to-Go), Branch Office, andBC4J clients. Restart your Mobile client after modifying the webtogo.ora file to enable tracing.

See Section B.3, "[DEBUG]" for a full description of the trace parameters. Each trace parameter matches the fields displayed on the General trace settings screen for the Mobile Server, as shown in Figure 16–1.

16.2.2 Turn on Tracing using the -d0 Option for Web-to-Go Clients With the WEBTOGO Executable

If you want to enable tracing quickly to a console window on the Web-to-Go Mobile client, execute the Mobile client webtogo command with the -d0 option. With the -d0 option, tracing is enabled and printed to a console window. The level of tracing shown is indicated by the TRACE_LEVEL parameter in the DEBUG section of the webtogo.ora file. All other DEBUG parameters are ignored in this situation. In order to start the Mobile client with the -d0 option, you must first stop your existing client.

You can only use this type of tracing for Mobile Client for the Web, Branch Office, and BC4J.

For more information on configuring the TRACE_LEVEL parameter in the webtogo.ora file, see Section B.3, "[DEBUG]".

16.2.3 View Device Logs

Each Mobile device maintains a log of the activity that it generates. See Section 8.3.7, "Viewing Device Logs" for more information.

16.3 Viewing the Log Files From the Application Server

Since Mobile Server uses OC4J as its application server, you can view the following log files to debug problems.

- Viewing OC4J server level output messages.

  ```
  <OC4J_HOME>\log\server.log
  ```

- Viewing HTTP requests handled by the server.

  ```
  <OC4J_HOME>\log\http-web-access.log
  ```

- Viewing exceptions or errors that are handled by OC4J.

  ```
  <OC4J_HOME>\application-deployments\webtogo\application.log
  ```

- Viewing 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
  ```

17

Reports

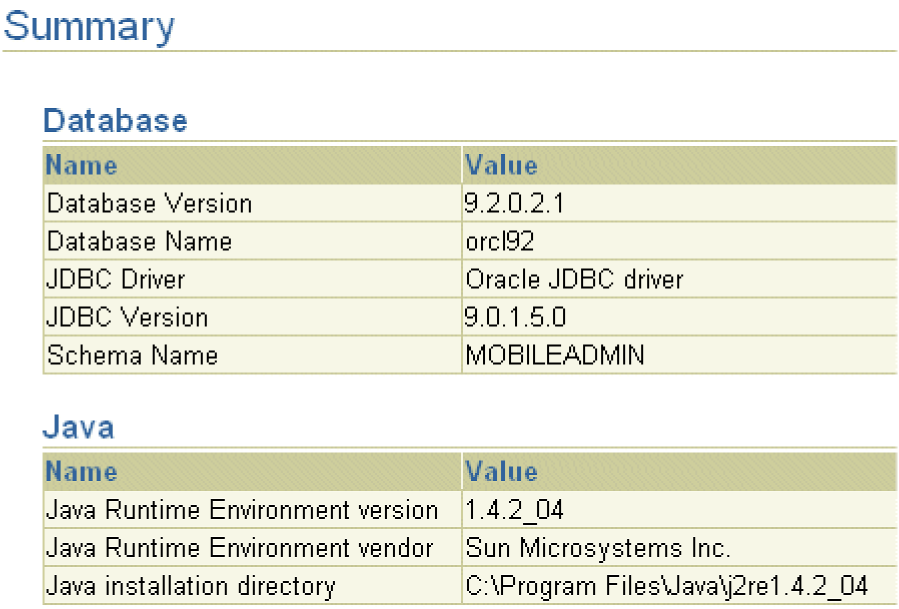
This chapter details the types of reports that you can view about the Mobile Server environment:

- Section 17.1, "Viewing System Status Reports for the Server"
- Section 17.2, "Viewing Active User Sessions"

17.1 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 17–1 displays, the Summary page lists Database, JRE, and Operating System details.

Figure 17–1 Summary Page



Summary

Database

Name	Value
Database Version	9.2.0.2.1
Database Name	orcl92
JDBC Driver	Oracle JDBC driver
JDBC Version	9.0.1.5.0
Schema Name	MOBILEADMIN

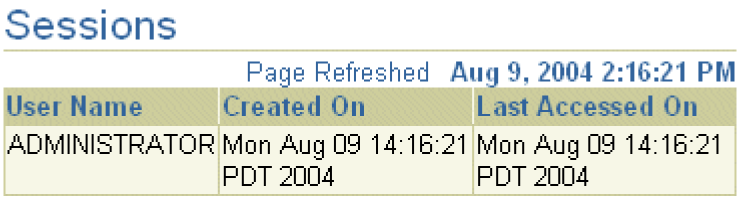
Java

Name	Value
Java Runtime Environment version	1.4.2_04
Java Runtime Environment vendor	Sun Microsystems Inc.
Java installation directory	C:\Program Files\Java\j2re1.4.2_04

17.2 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 17–2 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 17–2 Sessions Page



Sessions

Page Refreshed **Aug 9, 2004 2:16:21 PM**

User Name	Created On	Last Accessed On
ADMINISTRATOR	Mon Aug 09 14:16:21 PDT 2004	Mon Aug 09 14:16:21 PDT 2004

18

Adding Popular URLs as Bookmarks to Mobile Server Main Page

When you first bring up the Mobile Workspace, before you choose to go to the Mobile Manager, there is a Bookmark tab at the top of the page that lists popular URLs that an administrator has set up. These are URLs that are used often enough that you want to have them easily available.

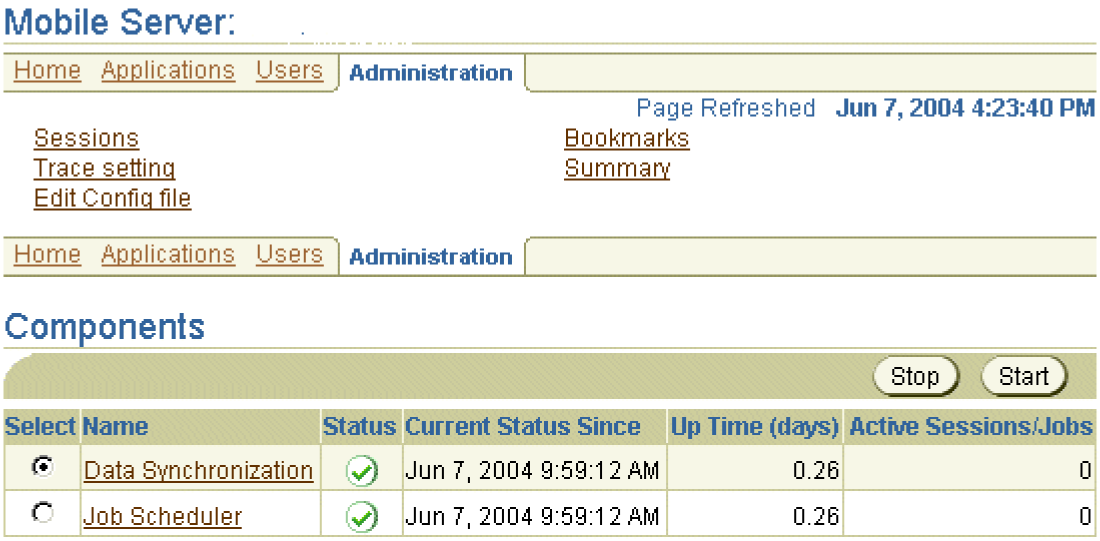
As an administrator, you can set up these bookmarks through the Administration page as detailed in the following sections:

- Section 18.1, "Setting Up Popular URLs as Bookmarks"
- Section 18.2, "Deleting Bookmarks"

18.1 Setting Up Popular URLs as Bookmarks

To add bookmarks to popular URLs, click **Administration**, as seen in Figure 18–1.

Figure 18–1 Administration Page



Mobile Server:

Page Refreshed **Jun 7, 2004 4:23:40 PM**

Sessions
Trace setting
Edit Config file

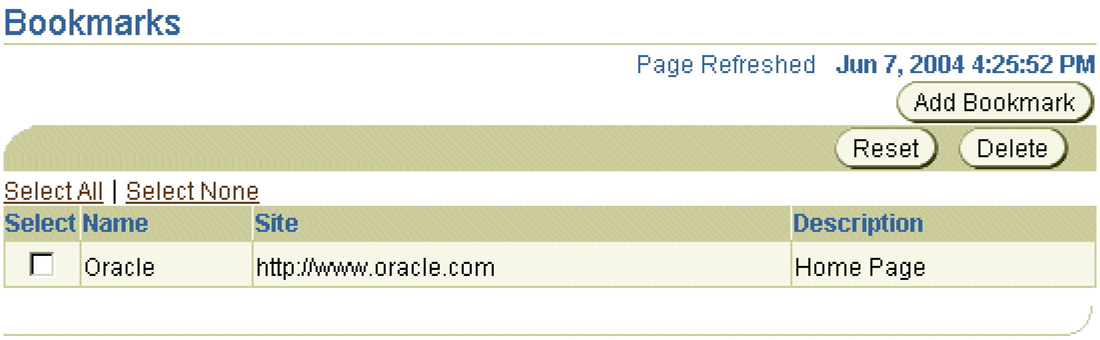
Bookmarks
Summary

Components

Select	Name	Status	Current Status Since	Up Time (days)	Active Sessions/Jobs
●	Data Synchronization	✓	Jun 7, 2004 9:59:12 AM	0.26	0
○	Job Scheduler	✓	Jun 7, 2004 9:59:12 AM	0.26	0

Click **Bookmarks**. As Figure 18–2 displays, the Bookmarks page appears.

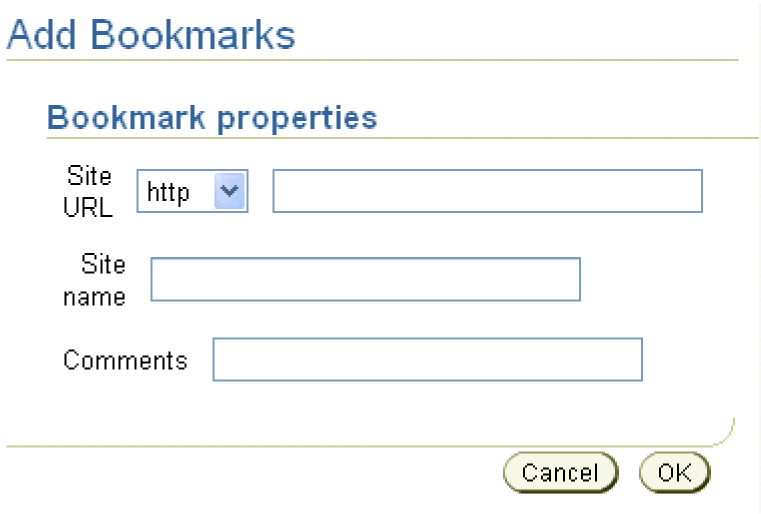
Figure 18–2 Bookmarks Page



Select	Name	Site	Description
□	Oracle	http://www.oracle.com	Home Page

Click **Add Bookmark**. As Figure 18–3 displays, the Add Bookmarks page appears.

Figure 18–3 Add Bookmarks Page



Enter data under the Bookmark Properties section as described in Table 18–1 and click **Save**. You are returned to the Mobile Server Bookmarks page which lists your bookmark.

Table 18–1 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

18.2 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**.

A
Troubleshooting FAQs

This document contains frequently asked questions for troubleshooting the Mobile Server. Topics include:

A.1 Synchronization Errors and Conflicts

Consult the following sections for details on how to resolve any synchronization errors or conflicts:

A.1.1 General Synchronization Errors and Conflicts

With the Mobile Server, you can have the following errors when synchronizing: nullity violations, foreign key constraint violations, or the client updates a row at the same time that the server deletes it.

The Mobile Server does not automatically resolve synchronization errors. Instead, the Mobile Server rolls back the corresponding transactions, and moves the transaction operations into the Mobile Server error queue. It is up to the administrator to view the error queue and determine if the correct action occurred. If not, the administrator must correct and re-execute the transaction. If it did execute correctly, then purge the transaction from the error queue.

A Mobile Server synchronization conflict occurs if:

- Nullity violations.
- Foreign key constraint violations.

- The client and the server update the same row.
- The client and server create rows with the same primary key values.
- The client deletes the same row that the server updates.
- The client updates a row at the same time that the server deletes it.

See Section 3.11, "Resolving Conflict Resolution with Winning Rules" in the *Oracle Database Lite Developer's Guide* for more information on conflict resolution techniques.

A.1.2 Synchronization Error if WinCE/Pocket PC Device Clock is Inaccurate

The WinCE Device clock must be accurate before attempting to synchronize. An inaccurate time may result in the following exception during synchronization: `CNS: 9026 "Wrong username or password. Please enter correct value and reSync."`

A.2 Problems When Synchronizing Large Number of Rows

When you synchronize a large number of rows, you may want to set the `AUTO_COMMIT_COUNT` parameter in the `POLITE.INI` file to a smaller number. The smaller the number is, the more often a commit occurs. If you do not set the `AUTO_COMMIT_COUNT` parameter to a smaller number, you may receive an `OutOfMemory` error.

See Section H.3.2.5, "AUTO_COMMIT_COUNT" for information on this parameter.

A.3 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 `-s` 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.4 Running the Mobile Server With Tracing Enabled

If you experience any difficulty with the Mobile Server running with the application server or the standalone Mobile Server, you can enable tracing in the Mobile Server.

To enable tracing in the Mobile Server, set up your environment as described in Section 16.1, "Enable Tracing on the Mobile Server". To enable tracing on your Mobile Client, follow the instructions in Section 16.2, "Enable Tracing on Mobile Clients".

A.5 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 the Microsoft Internet Explorer browser to time out.

The Solution

This solution is available only for the Microsoft 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.6 Accessing the Client Database Offline

When you go offline, after being online, and you need to access your client database, use your Mobile user 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 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.7 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*.

A.8 Troubleshooting An Out of Memory Error

The amount of memory that the JVM starts with is designated by the initial heap size option (`-Xms`) and the maximum heap size option (`-Xmx`) when starting the Mobile Server within the `runmobileserver` executable. The `runmobileserver` executable provides 256M for the initial heap size and 512M for the maximum heap size. In order to increase the amount of memory available to the JVM, try to allocate more memory to the Mobile Server when you start it. The administrator can allocate more memory to the JVM to handle the required load through the `-Xms` and `-Xmx` Java switches, as follows:

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

These switches explicitly set the amount of memory allocated for the initial and maximum Java heap size. The amount specified should be based on the available resources. The initial value should be equal to the maximum value to ensure the best performance. You should set the values to at least 256 MB. The following allocates 768M for both the initial and maximum heap sizes. Of course the amount of memory you allocate depends on what you have available.

```
java -Xms768m -Xmx768m -jar oc4j.jar
```

You may see an `OutOfMemory` exception during the client processing phase, which occurs when the client processes the `E` opcode sent from the server.

In addition, allocating more memory increases your performance. The memory requirements of the Mobile Server can vary dramatically across installations. The amount of memory that your applications use—for example, when propagating data, and so on—can exceed the amount of memory available in the JVM.

A.9 Troubleshooting an IllegalArgumentException

If you receive the `java.lang.IllegalArgumentException: Signal already used by VM:SIGINT` exception and you are using Branch Office, then you are using a JRE version previous to version 1.4.2. Update to JRE version 1.4.2 or later.

B

Configuration Parameters for the WEBTOGO.ORA File

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
`USE_SYSTEM_CLASSPATH=YES`	If set to yes, searches for Java classes in the computer's classpath before searching the Mobile Server Repository.
`MODE=SERVER`	The mode the Mobile Server is running in. Valid modes are `SERVER`, `CLIENT`, and `BRANCH`. The value `BRANCH` indicates that the Mobile Server is running in `BRANCH` mode for client operations.
`PORT=80`	The port number on which the Mobile Server is running. Not valid in Oracle9*i* Application Server (Oracle9*i*AS) installation.
`CLASSPATH=`	A list of directories in the Mobile Server Repository. Web-to-Go searches for Java classes.
`PROXY_SERVER=proxy.com`	The proxy host name and number. The Mobile Client for Web-to-Go setup modifies this entry.
`PROXY_PORT=80`	The proxy port number. The Mobile Client for Web-to-Go setup modifies this entry.

Table B–1 (Cont.) WEBTOGO Parameters

Parameter	Definition
SQL_RETRIES=5	Number of attempts to modify a JDBC connection before timing out.
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 Oracle9*i* Application Server (Oracle9*i*AS).
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:jdbcoracle: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.
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.
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 Oracle9*i* Application Server (Oracle9*i*AS).
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.

Table B–1 (Cont.) WEBTOGO Parameters

Parameter	Definition
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
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.
SERVER_URL=http://<mobile_server_name>:<port_number>/webtogo	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, https://<mobile_server_name>/setup, 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. 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 SERVER_URL parameter in the configuration file webtogo.ora, on the client side as displayed in the left column.

Table B–1 (Cont.) WEBTOGO Parameters

Parameter	Definition
SYNC_CANCEL	This parameter can be set on the client side to determine if the "Cancel" link should appear on the synchronization page. If this parameter is set to YES, 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.
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	This parameter must be set to the value YES only if the Mobile Server is running as a component of Oracle9*i*AS. Example: IAS_MODE=YES If the Mobile Server is running in Standalone mode or as a component of Oracle9*i*AS 1.0.2.2.0, this parameter must be set to the value NO. The default value is NO.
DISABLE_REMOTE_ACCESS	If set to YES, blocks a remote machine getting access to the Mobile Client for Web or Mobile Client for BC4J. You can set this parameter in the client side webtogo.ora file. Once this parameter is set and Mobile Client is restarted, only the request coming from the local machine is served by the Mobile Client listener. Any other request is blocked and not served. Default value is NO, which is necessary for Branch Office.

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	Type of File system. OL - Oracle Lite based file system. OS - Operating system’s file system. MIXED - Mixed file system.
PRIMARY=OL	Primary file system in MIXED mode.
SECONDARY=OS	Secondary file system in MIXED mode.
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.

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 Windows NT. Replace *`ORACLE_HOME`* with your actual home.

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`
`TRACE_DESTINATION`	Trace destinations are Console and File. The Administrator can set this parameter to any of these destinations. The Console option generates trace output to the console screen. **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. The File option generates trace output to a file. For more information, see `TRACE_FILE_NAME, TRACE_FILE_SIZE,` and `TRACE_FILE_POOL_SIZE`. Sample Value: `TRACE_DESTINATION=FILE`

Table B–3 (Cont.) DEBUG Parameters

Parameter Name	Definition

`TRACE_FILE_NAME=trace.log`	Used as base name to arrange trace files in sequential order starting from `1` to `FILE_TRACE_POOL_SIZE`.
	For example: If you set the following parameters.
	`TRACE_FILE_NAME=mytrace.log`
	`TRACE_FILE_POOL_COUNT=5`
	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.
	Sample Value: `trace.log`

`TRACE_LEVEL=1`	There are three levels of trace messages:
	`1(binary 00000001), Basic Trace`: General system information, most of the Web-To-Go trace output belongs to this level.
	`2 (binary 00000010), Function Trace`: Traces the function sequence being called, mostly used by Data Synchronization.
	`4 (binary 00000100), SQL Trace`: Traces SQL queries being executed, mostly used by Data Synchronization.
	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`.
	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.
	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.
	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.
	`3 & 1 (Basic) = 1 > 0`
	`3 & 2 (SQL)  =  2 > 0`
	`3 & 4 (Function) = 0 = 0`

Table B–3 (Cont.) DEBUG Parameters

Parameter Name	Definition
TRACE_USERS	List of valid user names. The user trace and system trace information which is listed is generated as trace output. 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. **Note**: As the administrator, you must not use the TRACE_NO_USER value as the user name. 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.
TRACE_FILE_PER_USER=YES	Used to specify an individual trace file pool for every individual user. Applicable only when the **File** option is the Trace destination. 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. If set to NO, all traceable users share the same trace file pool, the actual trace file does not contain any user name. Example: TRACE_FILE_POOL_PER_USER=No
TRACE_FILE_SIZE=10	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.
TRACE_FILE_POOL_SIZE=5	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

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>`	To call this public URL from your application, call it as follows:
	`http://<server>/public/<virtual path>`
	For example, `oracle.codeMyservlet=/my servlet`
	`oracle.codeMyservlet=/myservlet`

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 Data Synchronization. The values that are listed in the following table are default values.

Starting with the Oracle Database Lite 10*g* Edition, Data Synchronization uses a new log engine that supports new parameters for logging. They are:

- `GLOBALLogger`
- `SYNCLogger`
- `MGPLogger`
- `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.

Table B–7 (Cont.) Consolidator Parameters

Parameter Name	Definition
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
TRACE_LEVEL	Trace Level parameter can be set to the following trace message levels: MANDATORY: This option logs mandatory messages only. For example, Program Exceptions. Regardless of component settings, this option logs exceptions in the error log file (err.log) located in the Conslog directory. 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. NORMAL: This option logs normal messages that the user must be informed with and messages at the Mandatory and Warning level. INFO: This option logs information messages and messages at the Mandatory, Warning, and Normal levels. Examples: ■ Timing of synchronization: When the SYNCLogger is set to the TRACE_TYPE=TIMING and TRACE_LEVEL=INFO. ■ MGP Apply: When the MGPAPPLYLogger is set to the TRACE_TYPE=TIMING and TRACE_LEVEL=INFO. MGP Apply must be started with Timing of. COMPOSE must be started with Timing of MGP Compose. MGP must be started with Timing of. ■ COMPOSE: When the MGPAPPLYLogger is set to the TRACE_TYPE=TIMING and TRACE_LEVEL=INFO. ■ Status of MGP: When the MGPLogger is set to the TRACE_TYPE=GENERAL and TRACE_LEVEL=INFO. CONFIG: This option logs configuration messages and messages at the Mandatory, Warning, Normal, and Info levels. For example, JDBC driver version. FINEST: The finest level. This level is used for developers only. ALL: This option logs all messages according to the other settings such as Trace Type and Users.

Table B–7 (Cont.) Consolidator Parameters

Parameter Name	Definition
TRACE_TYPE	SQL: This option logs SQL-related messages only. For example, SQL statements. **Note**: This option is not trace level sensitive. 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. 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. 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. 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. GENERAL: This option logs messages that do not belong to any of the above listed trace types. **Note**: This type is trace level sensitive. ALL: This option generates logs of all trace types.
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.

Table B–7 (Cont.) Consolidator Parameters

Parameter Name	Definition
MAGIC_CHECK	Control the magic number checking of publication items. If enabled, and there is a mismatche between the server and the client magic numbers, then the publication item receives a complete refresh. If set to ALL, then the magic check is enabled, which is the default. If NONSHARED, then the magic check is enabled only for publication items that are not shared among users. A shared publication item has the following characteristics: ■ Read only ■ The publication item query either has no parameters or all users share the same parameter values. Setting to NONSHARED is useful when creating a installation CD for users that share data. See Section 9.8, "Creating a Single Package for Users That Share Data" for more information.

C Data Synchronization Requirements in INIT.ORA

The following sections describe the Data Synchronization requirements for Oracle and Oracle parameter settings in the `init.ora` file:

- 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 * PROCESSES + 5`
`TRANSACTIONS`	Default value: Derived: `(1.1 * SESSIONS)`
`DML_LOCKS`	Default Value: Derived: `(4 * 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, Data Synchronization 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 Data Synchronization 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, Data Synchronization needs 1500 DML locks. The default value for DML_LOCKS with PROCESSES set to 200 is 1000.

D

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 Name>
PASSWORD=<User Password>
```

```
ENCRYPTED=<True or False; True if the password is encrypted, False if not>
FULLNAME=<User Full Name>
PRIVILEGE=<User 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`** - Manage 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 name you want to add to this group>
USER=<User name you want to add to this group>
USER=<User 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 name you want to add to this group>
USER=<User 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 name you want to remove from this group>
USER=<User 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 name>
ACCESS=<Set access status as ENABLED>
ROLE=<Role of the group>
GROUP=<Groups 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 name>
ACCESS=<Access Status ENABLED/DISABLED>
ROLE=<Role of the group>
GROUP=<Group 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 name>
ACCESS=<Access Status>
ROLE=<Role of the group>
GROUP=<Groups 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 [DROPUSER] script, you can delete a user.

```
[DROPUSER]
NAME=<User Name>
```

Deleting a Group

Using the [DROPGROUP] script, you can delete a group.

```
[DROPGROUP]
NAME=<Group Name>
```

Deleting Access Privileges

Using the [DROPACL] script, you can delete access privileges provided to users.

```
[DROPACL]
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 name>
ACCESS=<Set access status as DISABLED>
ROLE=<Role of the group; set the value as DEFAULT ROLE or ADMINISTRATIVE ROLE>
GROUP=<Groups 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' contaning 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
```

E

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.

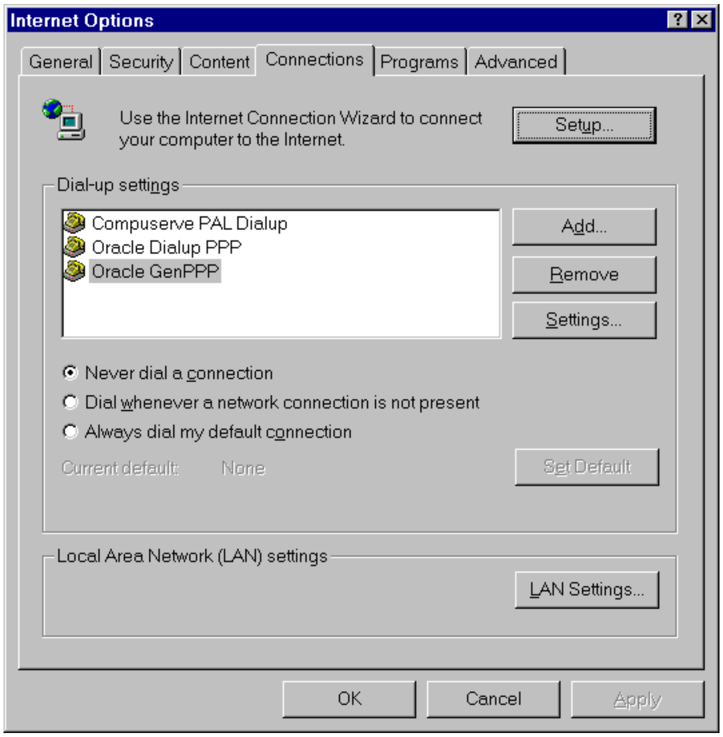
Administrators who usually connect to the network through a proxy server must bypass the proxy server before connecting to the Mobile Server.

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 E–1 displays the Connection tab.

Figure E–1 Connection Tab - Internet Explorer



Internet Options

Use the Internet Connection Wizard to connect your computer to the Internet.

Dial-up settings

- Compuserve PAL Dialup
- Oracle Dialup PPP
- Oracle GenPPP

● Never dial a connection

○ Dial whenever a network connection is not present

○ Always dial my default connection

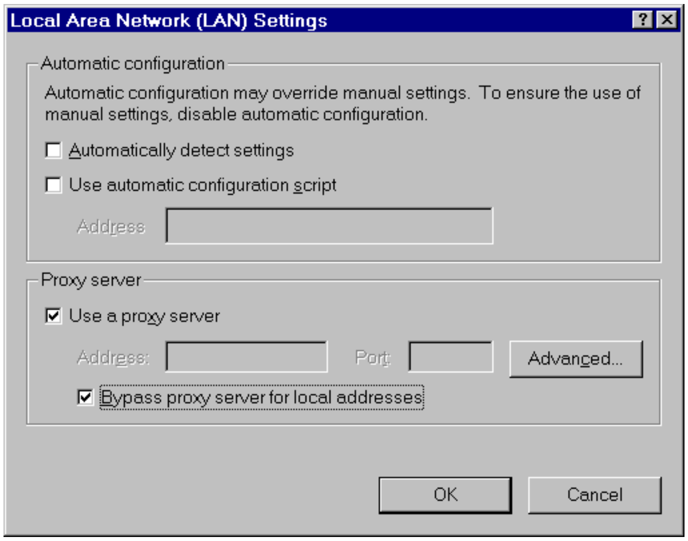
Current default: None

Local Area Network (LAN) settings

3. Click **LAN Settings**. Internet Explorer displays the Local Area (LAN) Network Settings dialog.

   Figure E–2 displays the Local Area (LAN) Network Settings dialog.

Figure E–2 Local Area Network Settings Dialog



Local Area Network (LAN) Settings

Automatic configuration

Automatic configuration may override manual settings. To ensure the use of manual settings, disable automatic configuration.

☐ Automatically detect settings

☐ Use automatic configuration script

Address:

Proxy server

☑ Use a proxy server

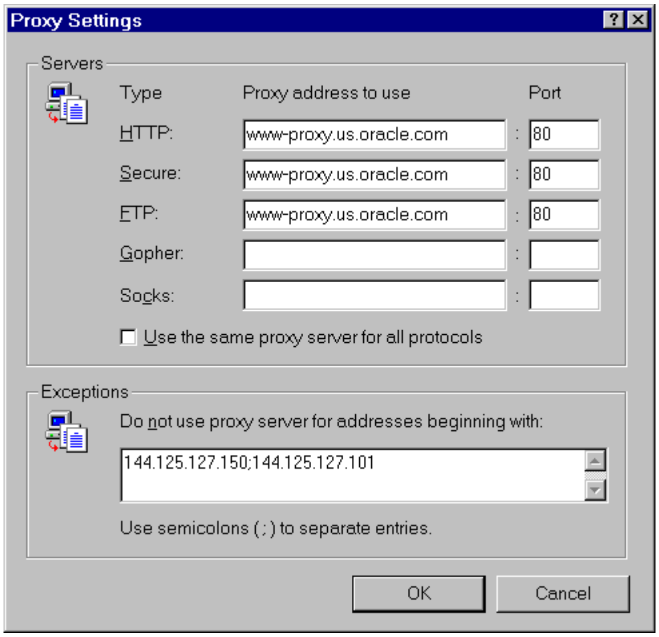
Address: Port:

☑ Bypass proxy server for local addresses

4. Select the **Bypass Proxy Server for Local Addresses** check box.
5. Click **Advanced**. The Proxy Settings dialog appears.

   Figure E–3 displays the Proxy Settings dialog.

Figure E–3 Proxy Settings Dialog



Proxy Settings

Servers

Type	Proxy address to use	Port
HTTP:	www-proxy.us.oracle.com	80
Secure:	www-proxy.us.oracle.com	80
FTP:	www-proxy.us.oracle.com	80
Gopher:		
Socks:		

☐ Use the same proxy server for all protocols

Exceptions

Do not use proxy server for addresses beginning with:

144.125.127.150;144.125.127.101

Use semicolons (;) to separate entries.

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**.

F

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.

F.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 F–1, " ALL_CLIENTS Parameters"
- Section F–2, " ALL_ERROR Parameters"
- Section F–3, " ALL_PUBLICATIONS Parameters"
- Section F–4, " ALL_SUBSCRIPTIONS Parameters"
- Section F–5, " ALL_SEQUENCES Parameters"
- Section F–6, " ALL_SEQUENCE_PARTITIONS Parameters"
- Section F–7, " ALL_PUBLICATION_ITEMS_ADDED Parameters"
- Section F–8, " ALL_PUBLICATION_ITEMS Parameters"
- Section F–9, " ALL_PUBLICATION_ITEM_INDEXES Parameters"
- Section F–10, " ALL_SUBSCRIPTION_PARAMS Parameters"

F.1.1 "MOBILEADMIN".CV$ALL_CLIENTS

The `"MOBILEADMIN".CV$ALL_CLIENTS` view provides information about Mobile Server clients.

Table F–1 provides a description of `ALL_CLIENT` parameters.

Table F–1 ALL_CLIENTS Parameters

Column	Datatype	Null	Description
CLIENT	VARCHAR(30)	NULL	The Mobile Server client
LASTREFRESH_STARTIME	VARCHAR(19)	NULL	Start time of the last refresh session
LASTREFRESH_ENDTIME	VARCHAR(19)	NULL	End time of the last refresh session

F.1.2 "MOBILEADMIN".CV$ALL_ERROR

The "MOBILEADMIN".CV$ALL_ERROR view provides information about failed client transactions.

Table F–2 provides a description of ALL_ERROR parameters.

Table F–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.

F.1.3 "MOBILEADMIN".CV$ALL_PUBLICATIONS

The ALL_PUBLICATIONS view provides information about Mobile Server publications.

Table F–3 provides a description of ALL_PUBLICATIONS parameters.

Table F–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.

F.1.4 "MOBILEADMIN".CV$ALL_SUBSCRIPTIONS

The ALL_SUBSCRIPTIONS view provides information about Mobile Server subscriptions.

Table F–4 provides a description of ALL_SUBSCRIPTION parameters.

Table F–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.

F.1.5 "MOBILEADMIN".CV$ALL_SEQUENCES

The ALL_SEQUENCES view provides information about Mobile Server sequences.

Table F–5 provides a description of ALL_SEQUENCES parameters.

Table F–5 ALL_SEQUENCES Parameters

Column	Datatype	Null	Description
NAME	VARCHAR2(30)	NULL	The sequence name.

F.1.6 "MOBILEADMIN".CV$ALL_SEQUENCE_PARTITIONS

The ALL_SEQUENCE_PARTITIONS view provides information about Mobile Server sequence partitions.

Table F–6 provides a description of ALL_SEQUENCE_PARTITIONS parameters.

Table F–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.

F.1.7 "MOBILEADMIN".CV$ALL_PUBLICATION_ITEMS_ADDED

The ALL_PUBLICATION_ITEMS_ADDED view provides information about Mobile Server publication items.

Table F–7 provides a description of ALL_PUBLICATION_ITEMS_ADDED parameters.

Table F–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. Note: You cannot use fast refresh synchronization with high priority restricting predicate.
WINNING_RULE	VARCHAR2(30)	NULL	The winning rules option for resolving replication conflicts. Options include "client wins" and "server wins".

F.1.8 "MOBILEADMIN".CV$ALL_PUBLICATION_ITEMS

The ALL_PUBLICATION_ITEMS view provides information about Mobile Server publication items.

Table F–8 provides a description of ALL_PUBLICATION_ITEMS parameters.

Table F–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. Note: You cannot use fast refresh synchronization with high priority restricting predicate.

F.1.9 "MOBILEADMIN".CV$ALL_PUBLICATION_ITEM_INDEXES

The ALL_PUBLICATION_ITEM_INDEXES view provides information about Mobile Server publication item indexes.

Table F–9 provides a description of ALL_PUBLICATION_ITEM_INDEXES parameters.

Table F–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.

F.1.10 "MOBILEADMIN".CV.$ALL_SUBSCRIPTION_PARAMS

The ALL_SUBSCRIPTION_PARAMS view provides information about Mobile Server subscription parameters.

Table F–10 provides a description of ALL_SUBSCRIPTION_PARAMS parameters.

Table F–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.

G

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 G.1, "Upload the Oracle JDBC Driver"
- Section G.2, "Change the Applet Tag"

G.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
```

G.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"
   ```

H

POLITE.INI Parameters

You can customize Oracle Database Lite by modifying the parameter values defined in your POLITE.INI file, which is available in Windows under %WINDIR%\POLITE.INI and in Linux under $OLITE_HOME/bin. You must have write permissions on the directory where this file is located to be able to modify the POLITE.INI file.

Note: On the WinCE and EPOC platforms, this file is named POLITE.TXT, so that you can double-click on it to open the file.

The following discusses the parameters in the different sections in the POLITE.INI file:

- Section H.1, "POLITE.INI File Overview"
- Section H.2, "All Databases Section"
- Section H.3, "Sync Client Parameters—SYNC Section"
- Chapter H.4, "Device Parameters—DMC Section"
- Section H.5, "Sample POLITE.INI File"

H.1 POLITE.INI File Overview

The POLITE.INI file centralizes database volume ID assignments, defines parameters for all databases on a system, and defines synchronization parameters. When you install Oracle Database Lite, the installation creates the POLITE.INI file in your Windows 98, NT, 2000, or XP home directory. On Windows CE and EPOC, the file name is POLITE.TXT.

The installation automatically sets the parameters in your POLITE.INI file, but you can modify them to customize the product behavior. To modify the POLITE.INI file, use an ASCII text editor.

H.2 All Databases Section

The following describes the parameters in the [All Databases] section of the POLITE.INI file.

- Section H.2.1, "CACHE_SIZE"
- Section H.2.2, "DATA_DIRECTORY"
- Section H.2.3, "DATABASE_ID"

- Section H.2.4, "DB_CHAR_ENCODING"
- Section H.2.5, "EXTERNAL_ENCRYPTION_DLL"
- Section H.2.6, "FLUSH_AFTER_WRITE"
- Section H.2.7, "MAX_INDEX_COLUMNS"
- Section H.2.8, "MESSAGE_FILE"
- Section H.2.9, "NLS_DATE_FORMAT"
- Section H.2.10, "NLS_LOCALE"
- Section H.2.11, "NLS_SORT"
- Section H.2.12, "OLITE_SERVER_LOG"
- Section H.2.13, "OLITE_SERVER_TRACE"
- Section H.2.14, "OLITE_SQL_TRACE"
- Section H.2.15, "SQLCOMPATIBILITY"
- Section H.2.16, "TEMP_DB"
- Section H.2.17, "TEMP_DIR"

H.2.1 CACHE_SIZE

Specifies the size of the object cache in kilobytes. The minimum is 128. If not set, the default is 4096 (4 megabytes).

H.2.2 DATA_DIRECTORY

On the WinCE platform, you may wish to define where the Oracle Lite database is installed. By default, the storage card is used—to preserve memory—and the storage card with the maximum free space is used. At least 32 MB of free space must be available. If there is not enough memory on the storage card, then the directory defaults to `\Orace`. If you want to specify the directory where the database is created, specify the directory in the `DATA_DIRECTORY` parameter, as follows:

```
DATA_DIRECTORY=\Orace
```

To synchronize, run `msync.exe`.

H.2.3 DATABASE_ID

Defines the next Database Volume ID number to be assigned the `CREATE DATABASE SQL` command. `DATABASE_ID` numbers must be unique for each database file on the system.

H.2.4 DB_CHAR_ENCODING

Specifies the Oracle Database Lite character set. If set to `NATIVE`, the default is the system default character set.

Table H–1 lists the supported code pages and their corresponding values of `DB_CHAR_ENCODING` for all supported languages.

Table H–1 Supported Code Pages and Values

Code Page	DB_CHAR_ ENCODING	Language
N/A	UTF8	All languages
(1250)	`ee8mswin1250`	(Croatian, Czech, Hungarian, Polish, Romanian, Slovak, and Slovenian)
(1251)	`cl8mswin1251`	(Bulgarian, Russian, and Ukranian)
(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))

H.2.5 EXTERNAL_ENCRYPTION_DLL

You can plug-in a custom encryption module for the Oracle Lite database by adding the `EXTERNAL_ENCRYPTION_DLL` parameter to the `POLITE.INI` configuration file. Use this if you do not want to use the default AES encryption provided for the Mobile client database.

You must either implement your encryption module into a DLL for the Windows environment or into a Shared Object (.SO) for the UNIX environment.

For example, if you created the encryption module as a DLL called `my_enc.dll`, which is located in the `C:\my_dir` directory, then you would add this module as the default encryption module in the `POLITE.INI` configuration file, as follows:

```
[All Databases]
EXTERNAL_ENCRYPTION_DLL=C:\my_dir\my_enc.dll
```

For more information, see the "Encryption Module" section in the Security chapter of the *Oracle Database Lite Developer's Guide*.

H.2.6 FLUSH_AFTER_WRITE

Syntax

```
FLUSH_AFTER_WRITE=TRUE|FALSE
```

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. 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.

H.2.7 MAX_INDEX_COLUMNS

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*.

H.2.8 MESSAGE_FILE

Use the MESSAGE_FILE parameter to specify the location of the message file used for the Mobile client Oracle Lite database. The default is where the binaries are installed. You may want to modify where the message file is located if you want to test another language. Modifying the MESSAGE_FILE parameter means that you do not have to move files around to test other languages.

Configure the path and the name of the message file, as follows:

```
MESSAGE_FILE=C:\Olite\Mobile\Sdk\BIN\OLITE40.MSB
```

H.2.9 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*.

H.2.9.1 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 H–2 lists date formats and their corresponding description.

Table H–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.

H.2.9.2 Date Format Examples

Listed below are sample variations of the NLS_DATE_FORMAT parameter.

1. `YYYY-MONTH-DAY:HH24:MI:P.M.`
2. `YYYY/MONTH/DD, HH24:MI A.M.`
3. `YYYY-MONTH-DAY:HH24:MI:PM`
4. `MM D, YYY, 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:SSSSS: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.`

H.2.10 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_FRANCE` in Oracle Database Lite. Table H–3 describes the supported locale and corresponding values of the `NLS_LOCALE` setting.

Table H–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`
Czech	`CZECH_CZECH REPUBLIC`
Danish	`DANISH_DENMARK`

Table H–3 (Cont.) Supported Locales and Values

Locale	NLS_LOCALE
Dutch (Netherlands)	`DUTCH_THE NETHERLANDS`
English (United Kingdom)	`ENGLISH_UNITED KINGDOM`
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`

H.2.11 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 the CREATEDB Section in the "Database Tools and Utilities For Win32 and WinCE Platforms" Appendix and the "Support for Linguistic Sort" Section in the "Oracle Lite RDBMS" Chapter in the *Oracle Database Lite Developer's Guide*.

H.2.12 OLITE_SERVER_LOG

The server log file contains the status of oldaemon processes including start, launch time, abort time, and executed processes. If any errors occurred, then the exception information is included. To forward all log information for a Multi-User Service on a LINUX machine, designate the filename of the logfile, as follows:

```
OLITE_SERVER_LOG = <path_and_filename>
```

H.2.13 OLITE_SERVER_TRACE

To debug the multi-user service, set this parameter to true, as follows:

```
OLITE_SERVER_TRACE = TRUE
```

See the "Debugging the Multi-User Service" section in the *Oracle Database Lite Getting Started Guide* for more information.

H.2.14 OLITE_SQL_TRACE

Generates the SQL statement text, compilation time, execution plan, and the bind value.

For example:

```
OLITE_SQL_TRACE = TRUE
```

SQL trace output is dumped to a trace file named oldb_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 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.

H.2.15 SQLCOMPATIBILITY

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.

H.2.16 TEMP_DB

The temporary database is created by default in virtual 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.

You can specify that the temporary database files are written to the file system either with the `TEMP_DB` or `TEMP_DIR` parameters. The `TEMP_DB` parameter enables you to define the name of the database files; the `TEMP_DIR` parameter allows you only to specify the directory to which the temporary database files are written.

To include this option, use the following syntax in the `POLITE.INI` file.

```
TEMP_DB=<path_and_temporary_database_name>
```

For example,

```
TEMP_DB=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, ...
```

H.2.17 TEMP_DIR

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. See Section H.2.16, "TEMP_DB" for more information.

H.2.18 SERVICE_PORT

Syntax

```
SERVICE_PORT=<port_number>
```

Default Value

```
1531
```

Modify the default port of the multi-user service with this parameter.

H.3 Sync Client Parameters—SYNC Section

Modify the `SYNC` section in the `POLITE.INI` file to control certain synchronization (OCAPI) functions. The following sections list the OCAPI parameters with their corresponding description and an 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*.

H.3.1 Overview of OCAPI—msync Client API

The msync 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. When you set the OCAPI parameters in the `POLITE.INI` file, then the parameter settings are implemented for the client on the first synchronization—based on the client platforms where the parameter settings need to apply.

An OCAPI function communicates with the Mobile Server through the selected transport and synchronizes the local database with the remote Mobile Server.

H.3.2 Synchronization Parameters

The following are synchronization parameters that you can modify:

- Section H.3.2.1, "TIME_LOG"
- Section H.3.2.2, "UPDATE_LOG"
- Section H.3.2.3, "COMPRESSION"
- Section H.3.2.4, "DEBUG"
- Section H.3.2.5, "AUTO_COMMIT_COUNT"
- Section H.3.2.6, "TEMP_DIR"
- Section H.3.2.7, "RESUME_CLIENT_TIMEOUT"
- Section H.3.2.8, "RESUME_CLIENT_MAXSEND"
- Section H.3.2.9, "ERROR_REPORT"
- Section H.3.2.10, "DB_ENCODING"
- Section H.3.2.11, "MEM_THRESHOLD"
- Section H.3.2.12, "VALIDATEDB"
- Section H.3.2.13, "ENCRYPT_DB"

H.3.2.1 TIME_LOG

Record the start and end time of a synchronization operation. 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. The administrator can 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.

H.3.2.2 UPDATE_LOG

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

H.3.2.3 COMPRESSION

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.

H.3.2.4 DEBUG

View debugging messages that are sent to the debug.txt file, which 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 debug.txt file. This enables OCAPI to invoke debugging messages.

Default Value

0

H.3.2.5 AUTO_COMMIT_COUNT

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

H.3.2.6 TEMP_DIR

Specify a directory for temporary files. OCAPI creates a temporary file for saving retrieved data. 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 system 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.

H.3.2.7 RESUME_CLIENT_TIMEOUT

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
```

H.3.2.8 RESUME_CLIENT_MAXSEND

Set the maximum data size in KiloBytes 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
```

H.3.2.9 ERROR_REPORT

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
```

H.3.2.10 DB_ENCODING

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.

H.3.2.11 MEM_THRESHOLD

Set memory threshold value in bytes for 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. OCAPI can prevent low memory conditions if you define the threshold correctly. If the available memory is lower than this value, OCAPI displays an error message.

Default Value

524288 (512 KB)

H.3.2.12 VALIDATEDB

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

H.3.2.13 ENCRYPT_DB

By default, the Oracle Lite database used by the Mobile client is not encrypted. However, you can ask for it to be encrypted through the ENCRYPT_DB parameter.

> **Note:** This parameter encrypts the database using the synchronization parameter.

- If set ENCRYPT_DB to 0, encryption is not executed. The database is left in whatever current state it is in.
- If set ENCRYPT_DB to 1, encryption of the database is executed only when a new Oracle Lite database (ODB) file is created. This is the preferred method if you want an encrypted database. Thus, the database is only encrypted when it is created.
- If set ENCRYPT_DB to 2, encryption of the database runs after every synchronization process. If you already have a database that is not encrypted, then you would want to set ENCRYPT_DB to 2, perform a synchronization—after which, the database is encrypted—and then set ENCRYPT_DB back to 1. This way, the database is encrypted, but is not encrypted after every synchronization, which would be a performance hit.

To encrypt an Oracle Lite database used by an embedded application, use the encrypb executable that is described in the "Oracle Lite Database Utilities" Appendix in the *Oracle Database Lite Developer's Guide*.

Default Value

0

H.4 Device Parameters—DMC Section

Do not modify any of the parameters in the DMC section. The only way to effect the values in this section is to modify an INF file for your device, as described in Chapter 8, "Manage Your Devices".

H.5 Sample POLITE.INI File

The following content is displayed from a sample POLITE.INI file.

```
[All Databases]
DATABASE_ID=128
DB_CHAR_ENCODING=NATIVE
CACHE_SIZE=4096
MAX_INDEX_COLUMNS=5
SQLCOMPATIBILITY=SQL92
NLS_DATE_FORMAT=RR/MM/DD H24,MI,SS
NLS_LOCALE=ENGLISH
TEMP_DB=c:\temp\olite_
TEMP_DIR=D:\TMP

[SYNC]
TIME_LOG=1
UPDATE_LOG=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