Oracle® Mobile Application Foundation

Implementation and Administration Guide

Release 11*i* (11.5.9)

Part No. A97695-04

April 2003

ORACLE®

Oracle Mobile Application Foundation Implementation and Administration Guide, Release 11*i* (11.5.9)

Part No. A97695-04

Contributing Authors: Ricardo Von Domburg, Anand Subbaraman, Matthew Self, Rob Van Attekum, Kannan Balakrishnan, Michelle Cheng

Contents

Send Us Your Comments v

Preface vii

Intended Audience vii
Documentation Accessibility vii
Related Documents vii

1 Introduction

1.1 Overview 1-1
1.2 About the Oracle Mobile Application Foundation 1-1
1.3 Migration 1-2

2 Technology Overview

2.1 Technology 2-1
2.2 Deployment 2-2

3 Dependencies

3.1 Oracle Mobile Application Foundation Dependencies 3-1
3.2 Installing and Configuring Oracle9i Lite 3-1

4 Implementation Tasks

4.1 Setting Profile Options 4-1
4.2 Publishing Mobile Applications 4-2

4.3 Scheduling Concurrent Programs 4-5

5 Performing Diagnostics and Troubleshooting

5.1 Running Diagnostic Tests 5-1
5.2 Configuring Debug Logging 5-4
5.3 Troubleshooting the Application Publishing 5-5
5.4 Troubleshooting the Mobile User Concurrent Program 5-5

6 Administering the Oracle Mobile Application Foundation

6.1 Configuring the Interapplication Bar 6-2
6.2 Manually Creating Mobile Users 6-2
6.3 Viewing User Synchronization Statistics 6-3
6.4 Working With Upload Errors 6-6
6.5 Managing Deferred Upload Errors 6-9
6.6 Managing Discarded Upload Errors 6-16
6.7 Managing Passed Upload Errors 6-18
6.8 Using the Oracle9i Lite Administration Screens 6-20

A Mobile Migration

A.1 Background A-1
A.2 Technology and Requirements A-2
A.3 Supported System Configurations A-2
A.4 Migration Overview A-2
A.5 Migration Tasks A-3

Send Us Your Comments

Oracle Mobile Application Foundation Implementation and Administration Guide, Release 11*i* (11.5.9)

Part No. A97695-04

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. 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?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can share comments with us by sending electronic mail to the following address: mobiledocs_us@oracle.com

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

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

Preface

Welcome to the *Oracle Mobile Application Foundation Implementation and Administration Guide*. This manual is intended to help you successfully implement and administer the Oracle Mobile Application Foundation product.

Intended Audience

This guide has been composed specifically for implementation engineers, system administrators, and any other administrator-level users who are required to set up and/or administer the Oracle Mobile Application Foundation.

Documentation Accessibility

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

Related Documents

Below is a list of the other documents that are referenced in this guide:

- *Installing Oracle Applications*

- *Oracle Mobile Field Service Implementation Guide for Laptops*
- *Oracle Mobile Field Service Implementation Guide for Palm Handhelds*
- *Oracle9i Lite Installation and Configuration Guide*
- *Oracle Applications CRM System Administrator's Guide*

1 Introduction

1.1 Overview

The Oracle Mobile Application Foundation provides a common infrastructure for the disconnected mobile applications that are offered in the Oracle E-Business Suite. In release 11.5.9, the Oracle Mobile Application Foundation supports Oracle Mobile Field Service applications. Please note that Oracle Mobile Field Service is available in two editions: Oracle Mobile Field Service/Laptop and Oracle Mobile Field Service/Palm.

1.2 About the Oracle Mobile Application Foundation

The Oracle Mobile Application Foundation replaces the Oracle CRM Gateway for Mobile Devices, which supported the Oracle Mobile Field Service applications up to and including release 11.5.7. In release 11.5.8 and later, the Oracle Mobile Application Foundation provides the infrastructure for disconnected mobile applications.

In the new architecture, the Mobile Gateway Database in the middle tier has been replaced by the Oracle9i Lite Mobile Server. Data replication is restricted to the enterprise and client tiers.

The advantages of the new mobile architecture include the following:

- Simple to install and maintain.
- Integrates with the CRM applications suite.
- Supports multiple platforms.
- More scalable.
- Simplifies user management.

- Supports real-time job assignment. (As soon as a job is assigned, it is available for downloading.)

1.3 Migration

If you have installed mobile products in previous 11i releases that use the Oracle CRM Gateway for Mobile Devices, please disregard Section 3.1, "Mobile Application Foundation Dependencies." Instead, go to Appendix A, "Mobile Migration." This appendix details the steps you must perform in order to successfully migrate your current mobile applications to the new architecture. If you are new to mobile applications, then you can ignore Appendix A.

Note: Oracle Field Sales/Laptop (ASL) and Oracle Field Sales/Palm (ASP) are no longer available as of Oracle Applications release 11.5.8. If you use these products, you must migrate to Sales Offline Edition. Please refer to the Oracle Sales Offline Edition User Guide on Oracle MetaLink.

2

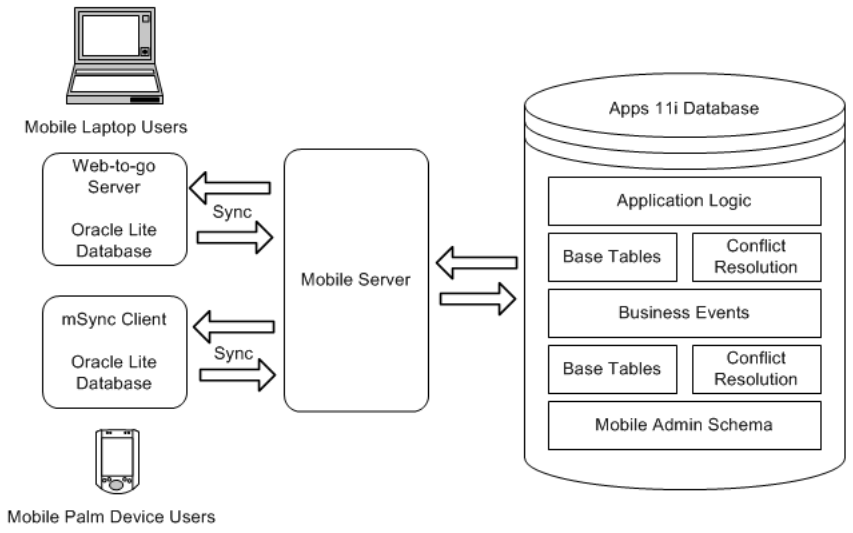
Technology Overview

2.1 Technology

The following components form the Oracle Mobile Application Foundation

- The Mobile Server Repository in the Applications database.
- The Oracle9i Lite Mobile Server.
- (If laptops are used) A local Oracle Lite database and a local Web-to-go server.
- (If Palm devices are used) A local Oracle Lite database and mSync clients on each Palm device.

The following illustration summarizes the relationships between these components. In the illustration, the Oracle Applications 11i database (including application logic, business events, mobile administration schema, base tables, and conflict resolution) resides on the enterprise tier. The enterprise tier is connected to the Mobile Server on the middle tier. Synchronizations are used to connect the Mobile Server on the middle tier to the components on the client tier. On the client tier, laptop clients have a local Oracle Lite database and a local Web-to-go server, while Palm clients have a local Oracle Lite database and mSync clients on each Palm device.



2.2 Deployment

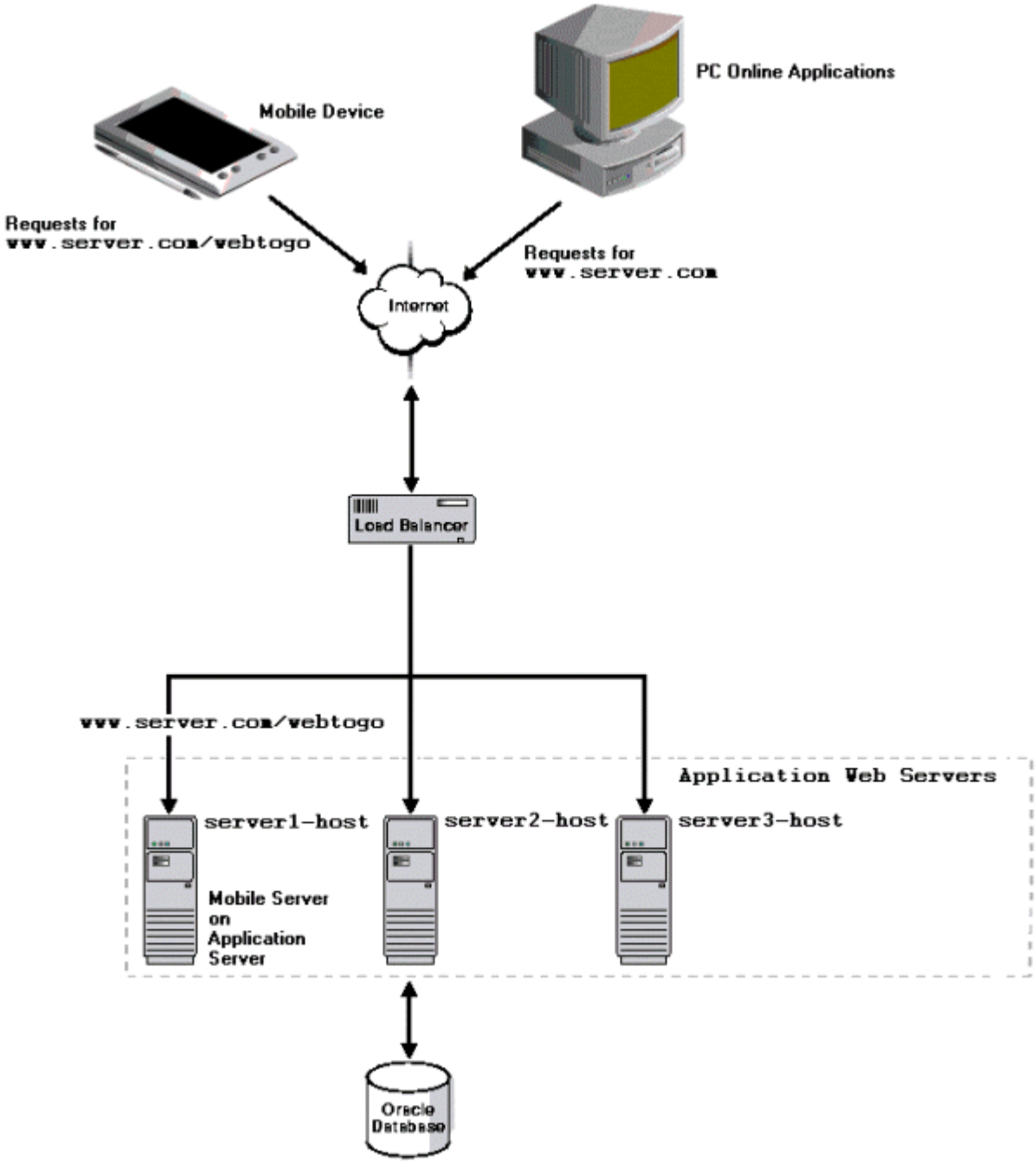
In the sample deployment illustrated below, there are three middle-tier application Web servers. "Server1-host" is dedicated to offline (mobile) users, while "server2-host" and "server3-host" are dedicated to online users.

The process flow begins on the client tier, where each user (whether mobile or online) opens an HTTP connection to the middle-tier servers through a load balancer/proxy. However, the offline users connect as:

```
http://<ip-address>:<port number>/webtogo
```

These URLs with "webtogo" are processed by the Mobile Server operating in standalone mode on "server1-host." URLs from the online PC applications (which do not contain "webtogo") are processed by an Apache/Jserv module on either "server2-host" or "server3-host."

Finally, the three Web servers on the middle tier connect to an Applications 11i database on the enterprise tier.


PC Online Applications
Mobile Device
Requests for
www.server.com/webtogo
Requests for
www.server.com
Internet
Load Balancer
www.server.com/webtogo
Application Web Servers
server1-host
server2-host
server3-host
Mobile Server
on
Application
Server
Oracle
Database

Note: The Oracle9i Lite Mobile Server is configured to run in standalone mode, independent of the Oracle9i Application Server or Apache installation. In this mode, Oracle9i Lite uses its own HTTP listener.

3

Dependencies

3.1 Oracle Mobile Application Foundation Dependencies

Before setting up the Oracle Mobile Application Foundation, there are several other Oracle applications or components which you must install and implement first. For details about these dependencies, please refer to Oracle*MetaLink* Note 226830.1.

3.2 Installing and Configuring Oracle9i Lite

You must install and configure Oracle9i Lite release 5.0.2 before implementing the Oracle Mobile Application Foundation. Oracle9i Lite is not included in the 11.5.9 Oracle Applications Rapid Install, so you must install Oracle9i Lite before proceeding to the Oracle Mobile Application Foundation implementation steps. For installation details, please see the appropriate platform-specific version of the *Oracle9i Lite Installation and Configuration Guide Release 5.0.2.*

If you are upgrading from a previous version of Oracle Applications, then you must apply the correct patch(es) for the Oracle Mobile Application Foundation before installing and configuring Oracle9i Lite.

The correct sequence is as follows:

1. If not using the 11.5.9 Rapid Install or 11.5.9 maintenance pack, then apply the 11.5.9 Mobile Application Foundation mini-pack.
2. Install and configure Oracle9i Lite release 5.0.2; then install Oracle9i Lite patch 5.0.2.3.
3. Implement the Oracle Mobile Application Foundation.

3.2.1 Installing and Configuring the Mobile Server

This section provides steps to install and configure Oracle9i Lite in standalone mode. This procedure installs and configures the Mobile Server on the application server that processes connections from mobile users.

3.2.1.1 Steps to Run the Environment Variable Script

Follow these steps to set all relevant environment variables to implement the Oracle Mobile Application Foundation.

1. Log in to the operating system as "applmgr" (the owner of the Applications file system) and set the environment. Execute the script APPSORA.env (UNIX) or envshell.cmd (Windows) to source the applications environment. For additional information, see "Setting the Environment" in the *AD Utilities Reference Guide*.
2. ***For UNIX***:
   a. Copy the file `$ASG_TOP/admin/template/asgovars_ux.env` to `$OAH_TOP/admin/scripts/asgovars.env`.
   b. Edit `$OAH_TOP/admin/scripts/asgovars.env` to set the values for the following variables:

      ```
      ORACLE_HOME = [location of the Oracle9i Lite installation]
      JAVA13_HOME = [location of the JDK 1.3.1 installation]
      TNS_ADMIN = [location of TNSnames.ora file]
      ```

   c. Execute the following command:

      ```
      . $OAH_TOP/admin/scripts/asgovars.env
      ```

   For Windows:

   a. Copy `$ASG_TOP\admin\template\asgoavars_nt.env` to `$OAH_TOP\admin\scripts\asgovars.cmd`.
   b. Edit `$OAH_TOP\admin\scripts\asgovars.cmd` to set the values for the following variables:

      ```
      set ORACLE_HOME=[location of the Oracle9i Lite installation]
      set JAVA13_HOME=[location of the JDK 1.3.1 installation]
      set TNS_ADMIN=[location of the TNSnames.ora file]
      ```

   c. Execute the following command:

      ```
      $OAH_TOP\admin\scripts\asgovars.cmd
      ```

Note: The script asgoavars.env (UNIX) or asgoavars.cmd (Windows) sets all environment variables for a mobile implementation.

3.2.1.2 Steps to Install Oracle9i Lite Mobile Server

Note: If you have already installed the Oracle database or iAS on this server, make sure to install Oracle9i Lite in a different directory. This will make the Oracle9i Lite Mobile Server run in standalone mode, without interfering with any other Oracle software installed on that server.

1. Use the Oracle9i Lite universal installer to install Oracle9i Lite. Follow the steps in the section "Installing the Mobile Server" of the appropriate platform-specific *Oracle9i Lite Installation and Configuration Guide Release 5.0.2* to install the Mobile Server.
2. Select **Mobile Server** in the **Installation Types** dialog box. This will configure the Mobile Server to use the Net9 libraries to connect to the database. Select **Yes** when asked if you want to install the repository.
3. In the **Mobile Server Repository Database Information** dialog box, enter the information for the Oracle Applications 11i database. This is where the Mobile Server Repository will be located. Please enter the service name and not the SID.
4. When prompted for the name of the schema for the Mobile Server Repository, enter **MOBILEADMIN**. You do not need to install the Mobile Development Kit or the SDK Demos.

   This completes the installation of the Mobile Server.
5. Edit the **webtogo.ora** file in ORACLE_HOME/mobile/server/bin for the port number. Port numbers less than or equal to 1023 will be able to run only as root.

For UNIX systems, you should use port numbers greater than 1023 if you will be starting the mobile server by users other than root.

6. To enable mobile user authentication from the Oracle Applications instance, edit the **webtogo.ora** file to contain the following lines:

   a. In the [webtogo] section, add:

   ```
   JAVA_OPTION= -DJAVA_COMPILER=NONE
   ```

   b. In the [EXTERNAL_AUTHENTICATION] section, add:

   ```
   CLASS=oracle.apps.asg.setup.AsgAuthenticator
   ```

   c. If using Windows, then add:

   ```
   USE_SYSTEM_CLASSPATH = YES
   ```

   Note: The option `-DJAVA_COMPILER=NONE` turns the Just In Time compiler (JIT) off. This is the recommended setting.

   Note: `[EXTERNAL_AUTHENTICATION]` is a literal string, not a variable.

   The **webtogo.ora** file should contain only one line specifying JAVA_OPTION. Additional parameters should be added to the same line and separated by a space.

7. Perform Step A below if you are using the 11.5.9 Rapid Install, or do not have apps.zip. Perform Step B below if you do have apps.zip.

   a. Make sure the Mobile Server can access the disc location referred to by the environment variable $JAVA_TOP.

   b. Append the location of the **apps.zip** file to the CLASSPATH. For example, an appropriate sample from the **webtogo** file would read as follows:

   ```
   CLASSPATH=.:${JAVA13_HOME}/jre/lib/rt.jar:${JAVA13_
   HOME}/jre/lib/i18n.jar:${JAVA13_HOME}/lib/tools.jar:${ORACLE_
   HOME}/mobile/classes/share.zip:${ORACLE_
   ```

```
HOME}/mobile/classes/jewt.zip:${ORACLE_
HOME}/mobile/classes/ojsp.jar:${ORACLE_
HOME}/mobile/classes/classgen.jar:${ORACLE_
HOME}/mobile/classes/javax-ssl-1_2.jar:${ORACLE_
HOME}/mobile/classes/jssl-1_2.jar:${ORACLE_
HOME}/mobile/classes/xmlparser.jar:${ORACLE_
HOME}/mobile/server/bin/webtogo.jar:${ORACLE_
HOME}/jdbc/lib/classes12.zip:${ORACLE_
HOME}/mobile/classes/consolidator.jar:${ORACLE_
HOME}/mobile/classes/servlet.jar:${ORACLE_
HOME}/mobile/classes/classgen.jar:/nfs/group/crmmob/code/mobxbld2/mobxbl
d2appl/java/apps.zip
```

Note: If the Mobile Server is installed on a tier other than the Applications Web tier, make sure the path to access the apps.zip file is valid.

For Windows systems, add the full path of apps.zip to the system classpath.

8. Start the Web-to-go listener by running the script **webtogo** from $ORACLE_HOME/mobile/server/bin.
9. Using a Web browser, connect to the Mobile Server by navigating to the following URL:

   http://<mobile_server_machine_name:port_number>/webtogo/startup

   Log in using the user ID **mobileadmin** and the associated password. (This is the password that you entered while installing Oracle9i Lite using the universal installer.)

   If the login is successful, then the **Menu Option** screen appears.
10. Click **Save** to save the encrypted versions of both the Mobile Server Repository user name and password in the webtogo.ora file. This also activates the Mobile Server Auto Start feature.

This completes the configuration of the Mobile Server. It now should be up and running.

3.2.2 Starting and Stopping the Mobile Server

Use the following procedure to start the Mobile Server:

Steps

1. Execute the script to set the Oracle Mobile Application Foundation environment variables as described in Section 3.2.1.1.
2. From $ORACLE_HOME/mobile/server/bin, run the script **webtogo.**

Use the following procedure to stop the Mobile Server:

Steps

1. Determine the process ID of the Web-to-go process by using the following command:

   ```
   ps -aef | grep webtogo
   ```

2. Issue a `kill -9` to stop the Mobile Server.

> **Note:** If you are using a Windows NT server, you can terminate the webtogo process from the Task Manager.

3.2.3 Installing the Oracle9i Lite Mobile Server on Additional Servers

You might want to set up the Mobile Server on additional servers to support a greater number of mobile users and distribute the load. In this case, the first server that you set up is your primary Mobile Server. Any additional servers are secondary servers. For more details on the definition of primary and secondary servers, please refer to Section 4.2, "Publishing Mobile Applications."

For every additional server you want to set up, follow the steps below:

1. Follow the steps in Section 3.2.1, "Installing and Configuring the Mobile Server" to install and configure the Mobile Server. When asked if you want to install the Repository, specify **NO**. The Mobile Server Repository has already installed as part of the primary server setup.
2. Edit **webtogo.ora** to point the parameter ROOT_DIR to the directory location of the Mobile Server Repository. You can copy this value from the webtogo.ora file of the primary Mobile Server instance that you created. Ensure that the location of this directory is accessible to the secondary Mobile Server(s) through NFS.

Note: In webtogo.ora, the parameter TRACE_FILE_POOL_SIZE is set to 5 by default. Under this default setting, only five users will synchronize when trace is enabled. Therefore, when trace is enabled, please set the parameter to a large number like 1000.

4

Implementation Tasks

This chapter contains instructions for the following procedures, which are required to successfully implement the Oracle Mobile Application Foundation. Perform these procedures after you have successfully installed and configured Oracle9i Lite as described in Section 3.2, "Installing and Configuring Oracle9i Lite."

- Setting Profile Options
- Publishing Mobile Applications
  - Understanding the Java Utility Command Syntax
  - Usage Examples
  - Publishing Applications with Multiple Mobile Server Instances
- Scheduling Concurrent Programs

4.1 Setting Profile Options

Set the following profile options for ASG and JTM. The table indicates the profile name, a brief description, the level at which to be set, and the recommended value (the default value).

Profile Option	Description	Level to be Set	Recommended (Default) Value
ASG_URL	The JDBC connection URL to the enterprise server.	Site	**jdbc:oracle:thin:@<HOST_ NAME>:<PORT_ NUMBER>:<ORACLE_SID> where HOST_NAME:PORT_ NUMBER:ORACLE_SID** is the database connect string to the enterprise database. This is a one time setup that should be changed only when the information about the enterprise database is changed, or when the enterprise database is copied and relocated to a different server.
ASG_CONS_PASSWD	The password for the Mobile Server Repository schema.	Site	Refer to Section 3.2, "Installing and Configuring Oracle9i Lite." This profile option is to be changed whenever the mobile server repository password is changed.
JTM: Mobile Applications Enabled	This setting dictates if the mobile applications will be used.	Application (Oracle Mobile Application Foundation)	**Y** if you want to enable mobile applications. **N** will disable updates to mobile devices.

4.2 Publishing Mobile Applications

For the mobile applications to function, you must create mobile application objects and publish the mobile application file. You can accomplish this by running the Java post-install program for each mobile application. The Java post-install program does the following:

- Registers the mobile object definitions for the mobile applications in the Oracle9i Lite Mobile Server Repository.
- Publishes the mobile applications to the Mobile Server Repository.

Use the following procedure to run the Java post-install program:

> **Note:** Please shut down the Mobile Server while publishing mobile applications. Restart the Mobile Server after publishing is completed.

Note: Before publishing the mobile applications, please go to the webtogo.ora file, and set the following parameters in the webtogo.ora file as shown:

- TRACE_ENABLED = YES
- TRACE_DESTINATION = CONSOLE
- TRACE_LEVEL = 7

These parameters appear twice in the webtogo.ora file and should be set identically in both places.

After the mobile applications have been published successfully, you should reset the above parameters to their original values.

Steps

1. Run the environment variable script as described in Section 3.2.1.1.
2. Next, publish the application for the mobile framework (JTM):

```
java -DJTFDBCFILE=<DBC FILE PATH>  -DAFLOG_ENABLED=TRUE
-DAFLOG_LEVEL=STATEMENT oracle.apps.asg.setup.InstallationMgr <DB_
HOSTNAME> <DB_PORTNUM> <ORACLE_SID> <APPS_USER_NAME> <APPS_PASSWD> JTM_
TOP JTM/JTM.ini html/download/jtm.zip
```

   Please see Section 4.2.1, "Understanding the Java Utility Command Syntax" for an explanation of the command syntax of this Java program.

3. Publish the mobile applications for the mobile products that you want to implement. Please see Section 4.2.2, "Usage Examples" for usage examples for Oracle Mobile Field Service/Laptop and Oracle Mobile Field Service/Palm.

4.2.1 Understanding the Java Utility Command Syntax

The following Java utility is used in Step 3 of the preceding procedure, "Publishing Mobile Applications":

```
java -DJTFDBCFILE=<DBC FILE PATH>  -DAFLOG_ENABLED=TRUE
-DAFLOG_LEVEL=STATEMENT oracle.apps.asg.setup.InstallationMgr <DB_HOSTNAME> <DB_
PORTNUM> <ORACLE_SID> <APPS_USER_NAME> <APPS_PASSWD> <PROD_TOP_NAME> <INI_FILE_
NAME> <ZIP_FILE_NAME>
```

Its command syntax is as follows:

- DJTFDBCFILE, DAFLOG_ENABLED, and DAFLOG_LEVEL are used by the JTT Logging Framework.
- DBC_FILE is usually found under the $FND_TOP/secure directory.
- The parameters <DB_HOST_NAME>, <DB_PORT_NUMBER>, <ORACLE_SID>,<APPS_USER_NAME>, and <APPS_PASSWD> are used to generate the connection string to the database.
- <PROD_TOP_NAME> refers to the application directory for the mobile product for which the publishing is done. It should be NULL if publishing is not done.
- <INI_FILE_NAME> refers to the name of the list file which has the list of files to be published. It should be NULL if publishing is not done.
- <ZIP_FILE_NAME> refers to the name of the zip file which contains the files to be published. The location of this file is relative to the<PROD_TOP_NAME>. It should be NULL if publishing is not done.

4.2.2 Usage Examples

In order to run the post-install step for Oracle Mobile Field Service/Palm, the command would be:

```
java -DJTFDBCFILE=<DBC FILE PATH>  -DAFLOG_ENABLED=TRUE
-DAFLOG_LEVEL=STATEMENT oracle.apps.asg.setup.InstallationMgr <DB_HOSTNAME> <DB_
PORTNUM> <ORACLE_SID> <APPS_USER_NAME> <APPS_PASSWD> CSM_TOP CSM/CSM.ini
html/download/CSM.zip
```

In order to run the post-install step for Oracle Mobile Field Service/Laptop, the command would be:

```
java -DJTFDBCFILE=<DBC FILE PATH>  -DAFLOG_ENABLED=TRUE
-DAFLOG_LEVEL=STATEMENT oracle.apps.asg.setup.InstallationMgr <DB_HOSTNAME> <DB_
PORTNUM> <ORACLE_SID> <APPS_USER_NAME> <APPS_PASSWD> CSL_TOP CSL/CSL.ini
html/download/csl.zip
```

4.2.3 Publishing Applications with Multiple Mobile Server Instances

If you plan to set up multiple Mobile Servers in order to support a greater number of mobile users and distribute the load, you should publish the mobile application only once, to the primary server.

The primary server is usually the Mobile Server which you set up first and installed the repository on. This server contains the directory location where the mobile application files are stored when you publish them. The webtogo.ora file contains an entry, ROOT_DIRECTORY, which points to this location.

Secondary mobile servers need to have access to the location of this root directory through network file sharing (NFS). The ROOT_DIRECTORY webtogo.ora entry of the secondary mobile servers will point to this directory location.

When you publish applications, you need to set the ORACLE_HOME to the location of the directory where the primary mobile server is installed.

4.3 Scheduling Concurrent Programs

Schedule the concurrent programs listed in the following table for the Oracle Mobile Application Foundation. For instructions on how to set up concurrent programs, refer to the *Oracle Applications System Administrator's Guide*.

Name	Description	Default (Recommended) Schedule
PROCESSES UPLOADED MOBILE DATA	Processes uploaded mobile data and applies it to the enterprise system.	Every 30 seconds.
MANAGE MOBILE USERS	Creates or removes mobile users automatically based on assigned mobile responsibilities.	Every 1 hour.
JTM MASTER CONCURRENT PROGRAM	The JTM master concurrent program, which runs all concurrent jobs registered by mobile applications.	TRANSACTION: every 5 mins. LOOKUP: every 30 mins. INVENTORY: once a day. PURGE: as often as required. Default is set to once per month.

5

Performing Diagnostics and Troubleshooting

This chapter describes how to run diagnostics and troubleshoot runtime problems for the Oracle Mobile Application Foundation. After your initial implementation, you should run the entire set of pre-seeded diagnostic tests to verify that the implementation has been successful. Subsequently the tests can be run whenever a problem arises.

5.1 Running Diagnostic Tests

Please refer to the *Oracle Applications CRM System Administrator's Guide* or the online help for the CRM System Administrator Console for instructions on how to use the Oracle Diagnostics for Applications user interface.

5.1.1 Running Pre-seeded Tests

Use the following procedure to run the pre-seeded Oracle Mobile Application Foundation diagnostic tests:

Steps

1. Log in as a system administrator to the CRM System Administrator Console.
2. Click the **Diagnostics** tab.
3. Choose **CRM Gateway for Mobile Services** from the **Application** drop-down list.
4. Click **Advanced**.
5. From here, you can run any of the following pre-seeded tests.

(The following table lists diagnostic tests, their descriptions, and associated symptoms and resolutions.)

Test	Description	Symptom	Resolution
Diagnostic check for Oracle9i Lite version	Checks if the Oracle9i Lite version installed is correct.	An error occurs while applying mobile application patches.	Review the diagnostic report and schedule or install/apply the correct Oracle9i Lite patches.
Diagnostic check for Oracle9i Lite Schema	Checks if the database access permissions for the Oracle9i Lite schema are set properly.	An error occurs while applying mobile application patches.	Running this diagnostic will examine and fix these problems automatically. Contact Oracle Support if you still see errors in this report.
Diagnostic case for profile option values	Checks if all the profile options required by the mobile foundation are set correctly.	An error occurs while applying mobile application patches.	Review the diagnostic report and set the appropriate profile option values.
Diagnostic case for upload concurrent job	Checks if the concurrent program that applies changes from mobile devices to the enterprise is working.	Changes from the mobile client are not applied to the enterprise system.	Review the diagnostic report and schedule or run the concurrent program to apply changes.
Diagnostic case for user concurrent job	Checks if the concurrent program to create, delete, and manage mobile users is working.	Users are not able to synchronize. They might get invalid username/password errors.	Review the diagnostic report and schedule or run the concurrent program to create mobile users.
Diagnostic case for JTM master concurrent program	Checks if the master concurrent program for running other mobile application concurrent programs is working	Many mobile application concurrent programs are not working.	Review the diagnostic report and schedule or run the concurrent program.
Diagnostic case for vertical user hook registration	Checks if mobile applications register vertical user hooks properly and the execute flag is set to run.	Changes on JTF Notes, CS Counter, or CSI Item Instance on the enterprise do not synchronize to the mobile device.	Reapply the JTM patch to register and set up the execute flag to run.
Diagnostic case for MAF environment variable verification	Checks if webtogo.ora and asgovars.env are set correctly.	Application publishes incorrectly or synchronization fails.	Change the scripts and republish the application.

5.1.2 Troubleshooting Mobile Operations Problems

Use this procedure to troubleshoot problems with mobile operations.

Prerequisites

Before running these diagnostics, you must:

- Configure the logger. For details, please see Section 5.2, "Configuring Debug Logging."
- Synchronize after configuring the logger.

Steps

1. Log in as a system administrator to the CRM System Administrator Console.
2. Click the **Diagnostics** tab.
3. Choose **CRM Gateway for Mobile Services** from the **Application** drop-down list.
4. Click **Advanced**.
5. As desired, run the tests below.

   (The following table lists diagnostic tests, their descriptions, and associated symptoms and resolutions.)

Test	Description	Symptom	Resolution
Logger for ASG creation/upgrade	Displays the log for the last mobile patch, including any changes to the mobile schema objects and publishing of mobile applications.	Users are not able synchronize and errors occur while applying mobile patches.	Contact Oracle Support and submit the log for further analysis.
Logger for user creation	Accepts a mobile user ID and displays the last user creation log for that user.	Users are not able to synchronize data.	Contact Oracle Support and submit the log for further analysis.
Logger for user synchronization	Accepts a mobile user ID and displays the log for the last user synchronization operation.	User synchronization errors occur, or the synchronization completes but users are not able to get data downloaded or uploaded to their devices.	Contact Oracle Support and submit the log for further analysis.

Test	Description	Symptom	Resolution
Error message for failed user creation	Displays the list of user IDs for whom the user creation or management operation failed.	Users are not able to synchronize.	Contact Oracle Support and submit the log for further analysis.

5.2 Configuring Debug Logging

Use this procedure to configure debug logging for troubleshooting purposes when you are experiencing problems with mobile operations. For further instructions on how to use the logging screens, please consult either the online help for the CRM System Administrator Console or the *Oracle Applications CRM System Administrator's Guide*.

Steps

1. Log in to the CRM System Administrator Console as system administrator.
2. Navigate to **Setting > System > Debug Logging > Configuration**.
3. Set the following values:
   - Enabled = **ON**
   - Level = **Statement**
   - Repository = **Database**
   - Module filter = **include string asg%, jtm%**
4. Click **Update** to save the values.
5. Bounce the Apache server on the middle tier.

Please note that logging at this level of detail consumes resources and could impact the performance of your system. Therefore, you should turn the logger off after you finish troubleshooting. This changes the values of the above fields back to their original values.

In addition to these logs, you can use the Oracle9i Lite Mobile Server logging facility to troubleshoot problems with user synchronizations. Please refer to the *Oracle9i Lite Mobile Server Administration Guide Release 5.0.2* for more details.

5.3 Troubleshooting the Application Publishing

Please check to ensure that the environment variables are set correctly in the file asgovars.env. Run the relevant diagnostic check as described in Section 5.1.1.

5.4 Troubleshooting the Mobile User Concurrent Program

This section discusses known issues regarding the concurrent program that creates mobile users, MANAGE MOBILE USERS.

Mobile users are created by this Java concurrent program. We require JDK 1.3.1 in the concurrent tier. Therefore you will need to apply patch 2356331 and follow the instructions in MetaLink Note 202128.1 in order to use JDK 1.3.1.

5.4.1 Problems and Resolutions

5.4.1.1 The Concurrent Program Fails to Create Mobile Users

Problem: After the patch is applied and the postinstallation setup is completed, the concurrent program still does not work.

Resolution: Restart the concurrent managers so that the new environment settings are initialized. To find out whether a concurrent request is being initialized to JDK 1.3.1 settings, use the following script:

```
select variable_name,value
  from fnd_env_context e,
       fnd_concurrent_processes p,
       fnd_concurrent_requests r
 where p.concurrent_process_id = e.concurrent_process_id
   and p.concurrent_process_id = r.controlling_manager
   and r.request_id=&request_id
   and e.variable_name in ('AF_JRE_TOP','AFJVAPRG','AF_CLASSPATH')
```

This script accepts a concurrent request ID and prints the environment settings for that request.

5.4.1.2 The Concurrent Program Only Works Intermittently

Problem: The concurrent program only works intermittently.

Resolution: Make sure **all** the concurrent managers are restarted. It is possible that only one concurrent manager was restarted. However, a concurrent request might be started by any of the concurrent managers. If all the concurrent managers are not

restarted, then some of the requests might not have the correct settings. To verify this, for requests that fail, use the script listed above.

5.4.1.3 The Concurrent Program Log Shows the Error Message "oracle.lite.web.resource.ResourceException: java/lang/ThreadLocal"

Problem: The log generated by the concurrent request shows this error.

Resolution: This is the clearest indication that the JDK version is not 1.3.1. The above mentioned class (ThreadLocal) is in JDK 1.2.2 or higher. Again, check if the environment settings are correct.

5.4.1.4 All Environment Settings Appear Correct, But the Concurrent Program Still Does Not Work.

Problem: The environment settings shown by the script listed in the above section "The Concurrent Program Fails to Create Mobile Users" indicate that JDK 1.3.1 is used. However, the concurrent program still fails.

Resolution: Look into the concurrent request's log file and look for any exceptions. Then for each of the environment variables AF_JRE_TOP, AFJVAPRG, and AF_CLASSPATH, check if the files listed exist.

6

Administering the Oracle Mobile Application Foundation

This chapter describes the administrative tasks that pertain to the Oracle Mobile Application Foundation:

- Configuring the Interapplication Bar
- Manually Creating Mobile Users
- Viewing User Synchronization Statistics
  - Viewing Synchronization Summary Tables on the Home Page
  - Viewing Synchronization Details on the Synch Detail Page
- Working With Upload Errors
  - Understanding Upload Errors
  - Purging Upload Errors
  - Viewing the Upload Errors Summary
- Managing Deferred Upload Errors
  - Working With the Deferred Upload Errors of a Particular User Synchronization Session
  - Editing the Details of Deferred Upload Errors
- Managing Discarded Upload Errors
- Managing Passed Upload Errors
- Using the Oracle9i Lite Administration Screens

Note: There is a mobile administrator responsibility that is similar to the system administrator responsibility. Any user who has been granted the mobile administrator responsibility can perform administrative activities for the Oracle Mobile Application Foundation. You must log in as a user with this responsibility to access the Mobile Foundation Administration Console, which is where the relevant administration screens are located.

6.1 Configuring the Interapplication Bar

If desired, you can set up the Interapplication Bar to conveniently navigate between the CRM System Administrator Console and the Mobile Application Foundation Administration Console. This is an optional procedure. Performing this setup will allow administrators to easily navigate from the Mobile Application Foundation Administration Console to the CRM System Administrator Console in order to run diagnostic tests and check error logs.

To navigate between the CRM System Administrator Console and the Mobile Application Foundation Administration Console, you will need to create a user who has both mobile administrator **("MobileAdmin")** and system administrator **("CRM HTML administration")** responsibilities. Log in as that user to the CRM System Administrator Console and set up the Interapplication Bar with the two applications being **"CRM Gateway for Mobile Services"** and **"CRM HTML administration"**.

Please refer to the *Oracle Applications CRM System Administrator's Guide* or the online help for the CRM System Administrator Console for instructions on how to successfully set up the Interapplication Bar.

6.2 Manually Creating Mobile Users

The Mobile Application Foundation allocates resources for mobile users when they are assigned a responsibility that is associated with a mobile application (such as the responsibility "Field Service/Laptop"). The concurrent program MANAGE MOBILE USERS runs periodically and creates or removes these mobile user resources based on the assigned responsibilities. A user who recently has been assigned a mobile responsibility will not be able to synchronize until after this

concurrent program runs. The user cannot change his or her password on the mobile client, and will need to do so online.

During system testing, it may be helpful to create users more rapidly. In this case, you can manually run the MANAGE MOBILE USERS concurrent program using the Concurrent Manager.

The home page of the Mobile Application Foundation Console displays the number of users who do not have the correct mobile subscriptions. In order to view the error messages associated with the mobile user subscription failures, please run the diagnostic test "Error Message for failed user creation." For details about how to run tests on the Mobile Application Foundation, please see Chapter 5, "Performing Diagnostics and Troubleshooting".

6.3 Viewing User Synchronization Statistics

This section covers the following tasks:

- Viewing Synchronization Summary Tables on the Home Page
- Viewing Synchronization Details on the Synch Detail Page

6.3.1 Viewing Synchronization Summary Tables on the Home Page

The home page of the Mobile Application Foundation Administration Console provides the following features:

- The **Critical Errors** table:

  This table displays a summary of critical synchronization and upload errors. You can click the links provided to display detailed error reports.

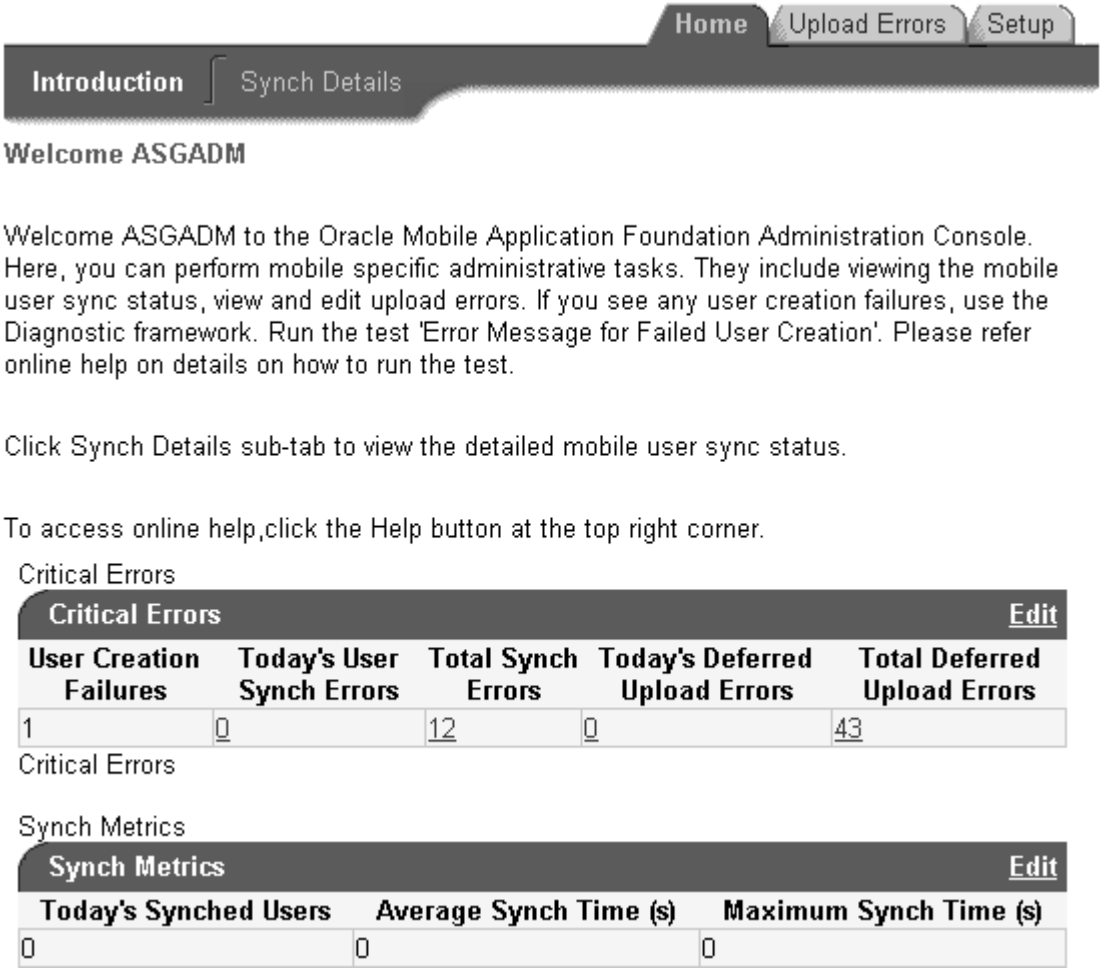
- The **Synch Metrics** table:

  This table displays a summary of synchronization metrics: You can view the number of synchronizations that occurred on the current date, the average time per synchronization, and the maximum synchronization time.

To navigate to the home page of the Mobile Application Foundation Administration Console, log in at the JTF login page as a user associated with the mobile administrator (**"mobileadmin")** responsibility. The home page will automatically display the two tables as described above.

Screenshot

An image of the Mobile Application Foundation Administration Console home page is provided below:



6.3.2 Viewing Synchronization Details on the Synch Detail Page

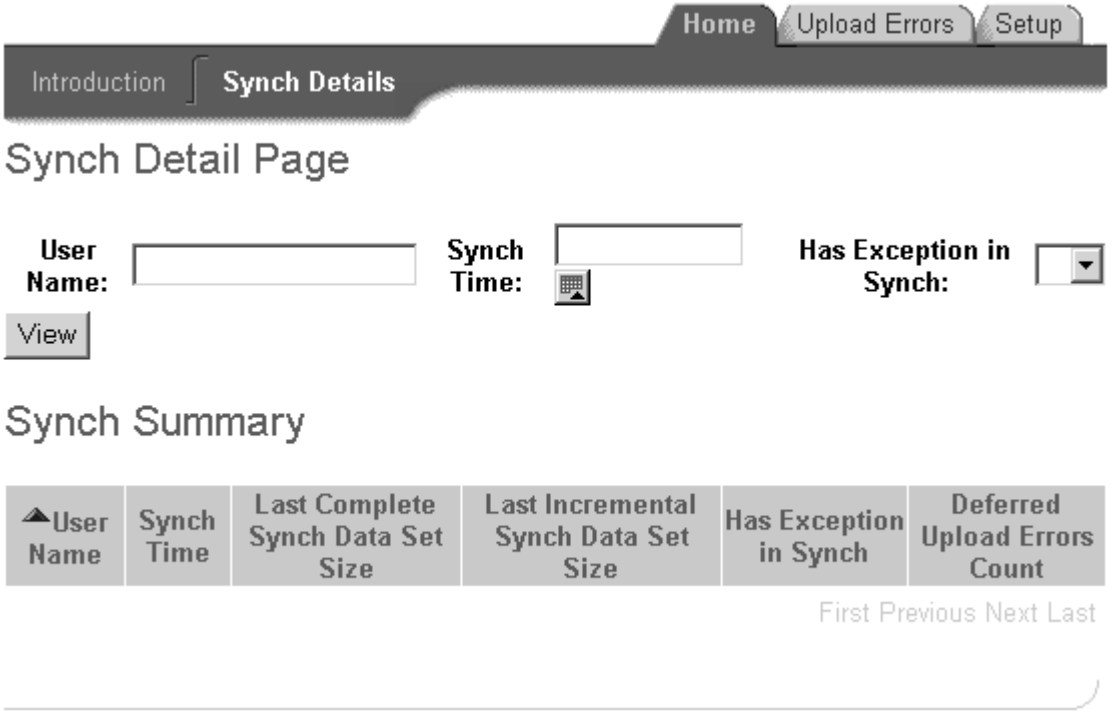
Use this procedure to view details about the synchronizations of particular user sessions.

Steps

1. In the Mobile Application Foundation Administration Console, from the **Home** tab, click the **Synch Details** subtab. On this page, the **Synch Detail Page,** the synchronization details for each user session is displayed.
2. Review the data provided, which includes user name, the time of the synchronization, the data set size of the last synchronization, if any exceptions occurred during the synchronization, and how many upload errors have occurred for that particular user. You can click the hyperlink in the **Deferred Transactions Count** column to view the details of the upload errors.
3. If desired, you can filter what data appears on this page. At the top of the page you can search by user name, synchronization time, and whether there was an exception in the synchronization. You can search by one or more of these parameters. To do so, enter values as desired into the fields, then click **View** to perform the search.

Screenshot

An image of the **Synch Detail Page** is provided below:



6.4 Working With Upload Errors

This section covers the following tasks:

- Understanding Upload Errors
- Purging Upload Errors
- Viewing the Upload Errors Summary

6.4.1 Understanding Upload Errors

During upload, some of the data that is synchronized might not be applied to the enterprise database.

The following situations might cause upload errors:

- **System Issues:** System issues might cause upload errors if the system is on a different mobile patch level than what is required. Administrators should verify the correct patch levels if this situation is suspected**.**
- **Setup Issues:** Setup issues might cause upload errors if the correct business process type or profile setting is not set on the enterprise system. For example, in an expense line, it is necessary that the expense type is set correctly. If the expense type is set as "lunch" on the enterprise system while on the mobile device the expense line is set as "dinner," then this issue might cause an upload error.
- **Faulty Data or Data Conflicts:** Faulty data or data conflicts can also cause upload errors. For example, multiple people attempting to update the same record can cause data conflicts where the server and the client to go out of sync.

The Mobile Application Foundation maintains a log of all errors that occur during the upload process. In the Upload Error pages, you can view and reapply these errors. In the case of a system or setup issue, you only need to fix the system or setup issue and then reapply the error. In the case of faulty data or data conflicts, you need to modify the data and then apply the error.

You can change data or system elements as needed and then reapply errors to the enterprise database. If they are successful, then you can delete them from the error tables. If they continue to fail, then you can choose to either modify them once more and try again, or to discard the failed errors and delete them from the error tables.

The Mobile Application Foundation logs every user synchronization session that has errors during upload, as well as the actual errors. For example, a user synchronization session could consist of uploading two service requests and three debrief lines. In this session, suppose two of the debrief lines failed to be applied to the enterprise system. This session would have two upload errors. You can choose to discard one or both of these errors. However, when you reapply even one of the errors, all the errors in the user session will be reapplied. Therefore, it is recommended that you make changes to all the errors in a synchronization session before reapplying the session.

Depending on how they are handled by the administrator, upload errors can be classified as follows:

- **Deferred**: These transactions did not go through to the main database tables. Either they have never been reapplied, or they have been reapplied multiple times and always failed.
- **Passed**: These transactions have been successfully reapplied. Only transactions that originally failed and later were successfully reapplied are shown here.

Passed upload errors can be purged (permanently deleted) from the system by an administrator.

- **Discarded**: These are transactions that you want to remove from the system because reapplying them has never been successful. Discarded upload errors are analogous to files that are in the "Recycle Bin" or the "Trash." Discarded upload errors can be purged (permanently deleted) from the system by an administrator.

6.4.2 Purging Upload Errors

Administrators should periodically purge all stored upload errors, because the table in the enterprise system which holds them will grow infinitely large unless its contents are occasionally removed.

For instructions on how to purge discarded upload errors, please see Section 6.6, "Managing Discarded Upload Errors." For instructions on how to purge passed upload errors, please see Section 6.7, "Managing Passed Upload Errors."

6.4.3 Viewing the Upload Errors Summary

Use this procedure to view the upload errors summary.

Steps

1. In the Mobile Application Foundation Administration Console, navigate to the **Upload Errors** tab.
2. The **Introduction** page provides summary information for deferred, discarded, and passed upload errors. To view further information about the errors, click the appropriate **Detail** button. Alternatively, you can click a subtab to navigate to the respective detail page for that error type.

Screenshot

An image of the Upload Errors **Introduction** page is provided below:

Introduction

The upload error screens allow the user to view and reapply upload errors.

These errors can occur during upward sync. You can view, edit and reapply any deferred upload errors. Passed upload errors are errors that have been reapplied successfully. They can be purged. Discarded are those upload errors that fail repeatedly and can be purged.

To access online help,click the Help button at the top right corner.

Upload Errors	Sessions	Total Errors	Detail
Deferred	43	67	Detail
Discarded	17	19	Detail
Passed	0	0	Detail
Total Errors		86	

6.5 Managing Deferred Upload Errors

Deferred upload errors are transactions that did not go through to the main database tables. Either they have never been reapplied, or they have been reapplied multiple times and always failed. Use this procedure to manage deferred upload errors.

Steps

1. In the Mobile Application Foundation Administration Console, choose the **Upload Errors** tab and then the **Deferred** subtab to navigate to the main **Deferred Upload Errors** page.
2. If desired, you can search for errors by one or more of the provided parameters: user name, synchronization date, publication name, publication item name, or the name of the wrapper that failed. The default result set displays all user synchronization sessions that failed. To perform a search, enter values into the appropriate fields, then click **View**. You can use the percent sign (%) as a wildcard character.
3. Review the **Deferred Upload Errors Summary** table, which shows a summary of user synchronization session errors. The table contains the following columns:
   - **User Name**: The name of the mobile user whose data had errors.
   - **Synch ID**: The ID of the synchronization session.
   - **Last Apply Date**: The date and time at which the user last synchronized the data.
   - **Error Count**: The number of errors that have occurred.
   - **Times Applied:** This displays the number of times the transaction was reapplied unsuccessfully, after it was deferred. For a new upload error, the value is "1."
   - **Select:** In this column you can select a check box to work with a particular error (you can reapply it, discard it, or view its details).
4. To view further details about a particular user synchronization session, click the **Details** button located in that row in the table.
5. To discard errors, select the appropriate check box(es), then click the **Discard** button.
6. To reapply errors, select the appropriate check box(es), then click the **Reapply** button.

Screenshot

An image of the main **Deferred Upload Errors** page is provided below:


Home
Upload Errors
Setup
Introduction
Deferred
Discarded
Passed
Deferred Upload Definition
User Name: %
Synch Date:
Publication Item Name: %
Wrapper Name %
View
Deferred Upload Summary
User Name Synch ID Last Apply Date Error Count Times Applied Select Detail
RSREDDY 403 28-FEB-2003 02:18:10 2 1 Detail
RSREDDY 401 28-FEB-2003 02:13:08 1 1 Detail
RSREDDY 400 28-FEB-2003 02:13:07 1 1 Detail
RSREDDY 395 28-FEB-2003 01:32:56 1 1 Detail
RSREDDY 393 28-FEB-2003 01:17:33 1 1 Detail
RSREDDY 391 28-FEB-2003 01:02:38 1 1 Detail
RSREDDY 389 28-FEB-2003 00:55:03 1 1 Detail
RSREDDY 386 27-FEB-2003 23:54:36 1 1 Detail
RSREDDY 382 27-FEB-2003 23:29:20 1 1 Detail
RSREDDY 379 27-FEB-2003 23:14:10 1 1 Detail
First Previous 1 - 10 of 43 Next Last
Discard
Reapply
Select All
Clear All

6.5.1 Working With the Deferred Upload Errors of a Particular User Synchronization Session

Use this procedure to work with the deferred upload errors of a particular user synchronization session.

Steps

1. To reach the **Deferred Upload Errors Detail** page, follow this navigation path in the Mobile Application Foundation Administration Console:
   a. Go to the **Upload Errors** tab.
   b. Go to the **Deferred** subtab.
   c. Click a **Detail** button.

   This page displays information about the user synchronization session and the errors that occurred in that session. The error information displayed includes the following:

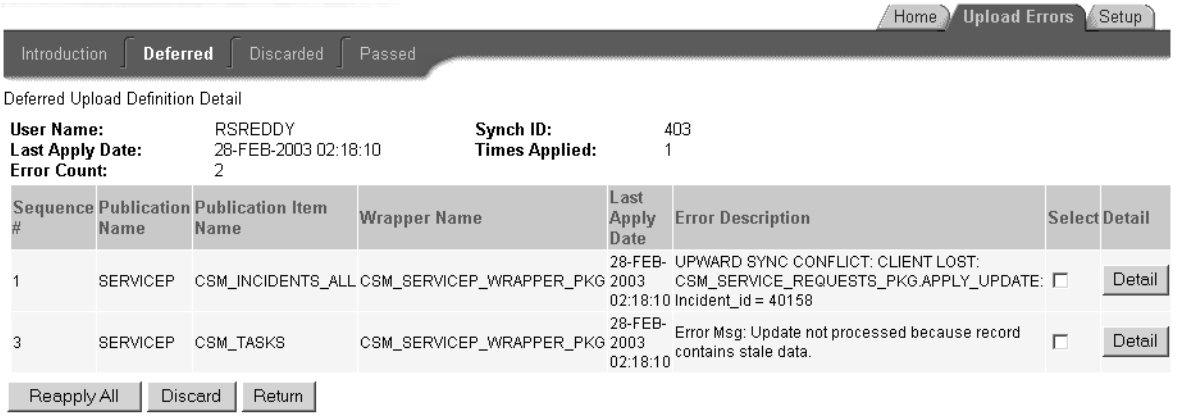
   - **Sequence Number**: A synchronization session includes all the records updated by the mobile user. The sequence number indicates the order of this record among the records that were applied to the enterprise database.
   - **Publication Name**: Publication name refers to a group of mobile tables which comprise the data store for the mobile application in the mobile device. For example, SERVICEL refers to Oracle Mobile Field Service/Laptop, while SERVICEP refers to Oracle Mobile Field Service/Palm. A mobile application is composed of Publications (data) and a mobile application (code).
   - **Publication Item Name**: Publication item names are the names of the mobile tables in the Laptop or in the Palm. A Publication Item is part of a Publication.
   - **Wrapper Name**: The wrapper name is the name of the PL/SQL packaged procedure which is responsible for applying the client changes to the enterprise database.
   - **Last Apply Date**: The date when the error was last applied.
   - **Error Description**

2. To discard one or more errors, select the appropriate check box(es) in the table and then click **Discard**.

3. To reapply the errors, click **Reapply All**. All the errors in this particular synchronization session will be reapplied.
4. To access the page where you can view and edit the details of a particular error, click the appropriate **Detail** button.

Screenshot

An image of the **Deferred Upload Errors Detail** page is provided below:



6.5.2 Editing the Details of Deferred Upload Errors

Use this procedure to edit the details of a particular deferred upload error.

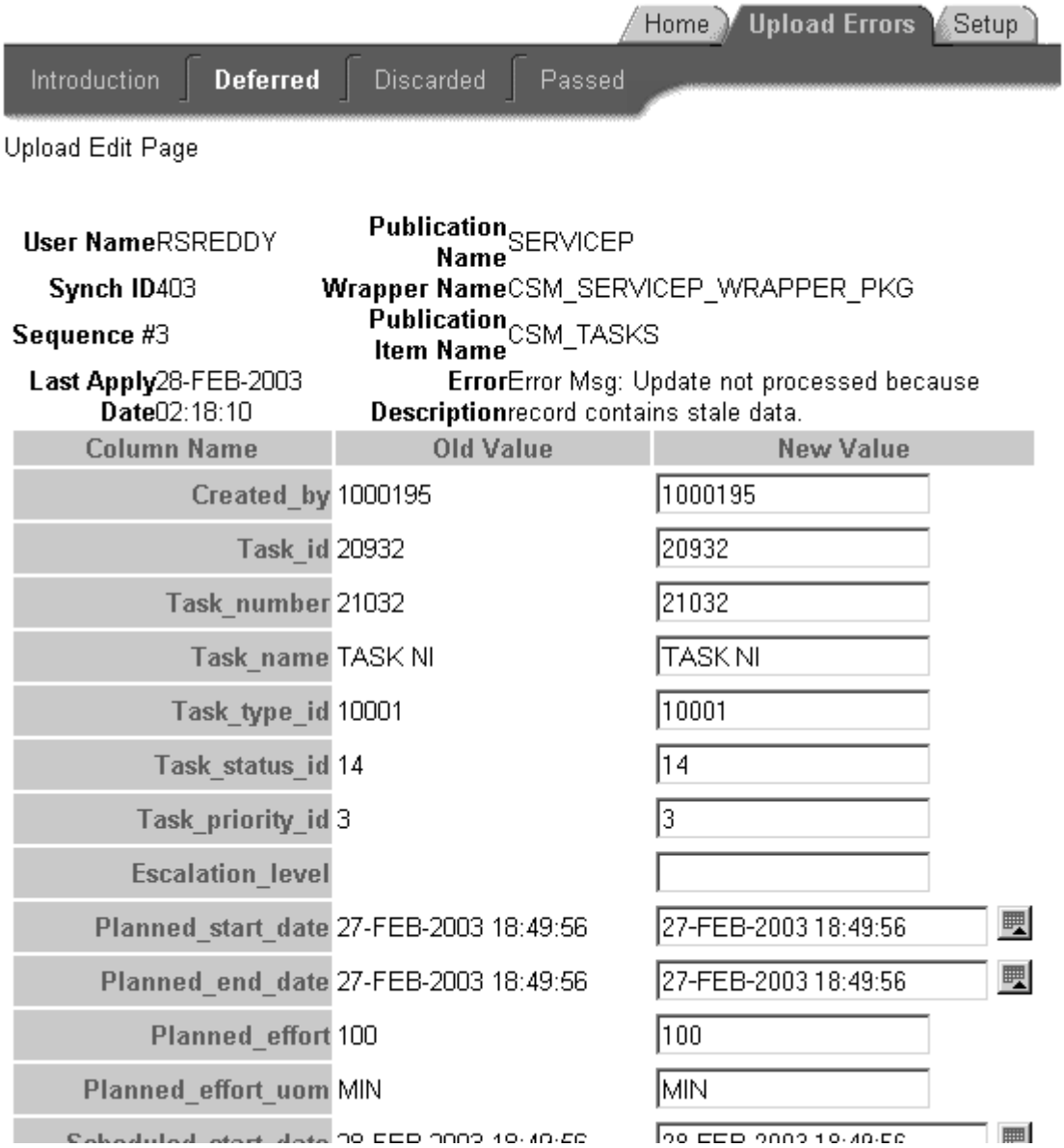
Steps

1. To reach the page where you can edit the details of deferred upload errors, follow this navigation path in the Mobile Application Foundation Administration Console:
   a. Go to the **Upload Errors** tab.
   b. Go to the **Deferred** subtab.
   c. Click a **Detail** button in the table, opening the **Deferred Upload Errors Detail** page.
   d. On that page, click a **Detail** button in the table.
2. The **Deferred Upload Errors Edit Page** page displays details about a particular deferred upload error. On this page, you can do one of the following:

- Discard this error: Click **Discard**.
- Change the values of this error: If faulty values are causing the upload error, you can edit values as desired in the fields provided, and then click **Save** to apply changes.
- Do nothing and return to the **Deferred Upload Errors Detail** page: Click **Return**.

Screenshot

An image of the **Deferred Upload Errors Edit Page** is provided below:


Home
Upload Errors
Setup
Introduction
Deferred
Discarded
Passed
Upload Edit Page
User Name RSREDDY
Publication Name SERVICEP
Synch ID 403
Wrapper Name CSM_SERVICEP_WRAPPER_PKG
Sequence # 3
Publication Item Name CSM_TASKS
Last Apply Date 28-FEB-2003 02:18:10
Error Description Error Msg: Update not processed because record contains stale data.
Column Name
Old Value
New Value
Created_by 1000195 1000195
Task_id 20932 20932
Task_number 21032 21032
Task_name TASK NI TASK NI
Task_type_id 10001 10001
Task_status_id 14 14
Task_priority_id 3 3
Escalation_level
Planned_start_date 27-FEB-2003 18:49:56 27-FEB-2003 18:49:56
Planned_end_date 27-FEB-2003 18:49:56 27-FEB-2003 18:49:56
Planned_effort 100 100
Planned_effort_uom MIN MIN

6.6 Managing Discarded Upload Errors

Discarded upload errors are transactions that you want to remove from the system. Reasons for discarding transactions include the following:

- Reapplying them has never been successful.
- The mobile user entered an incorrect value and wants to undo the transaction.
- The transaction has become old due to multiple failures and the user has created more recent transactions.

Discarded errors are analogous to files in the "Recycle Bin" or the "Trash." Use this procedure to manage discarded upload errors.

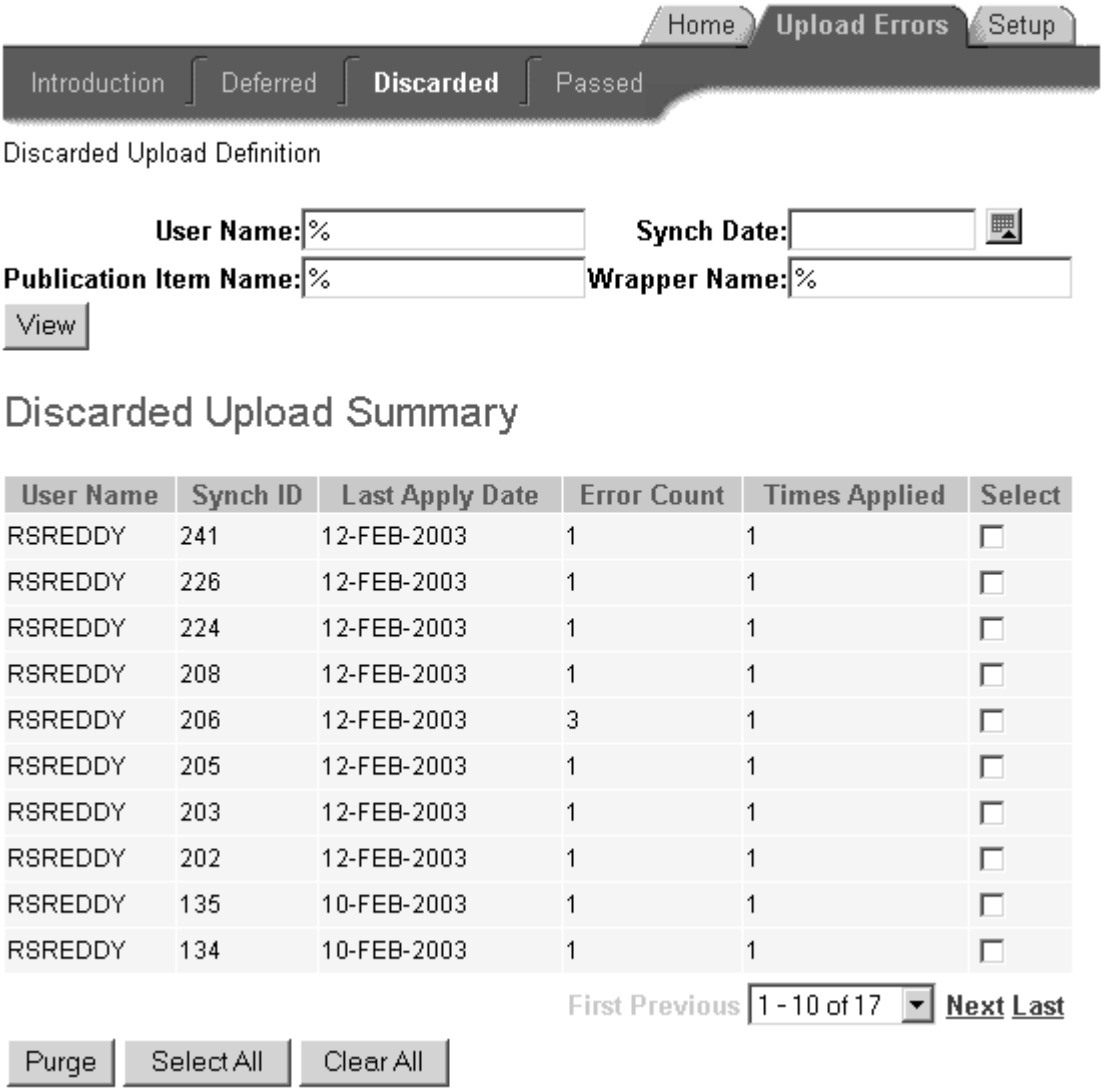
Steps

1. In the Mobile Application Foundation Administration Console, choose the **Upload Errors** tab and then the **Discarded** subtab to navigate to the **Discarded Upload Errors** page.
2. If desired, you can search for errors by one or more of the provided parameters: user name, synchronization date, publication name, publication item name, or the name of the wrapper that failed. The default result set displays all user synchronization sessions that have been discarded. To perform a search, enter values into the appropriate fields, then click **View**. You can use the percent sign (%) as a wildcard character.
3. Review the **Discarded Upload Errors Summary** table, which lists discarded user synchronization session errors. The table contains the following columns:
   - **User Name**: The name of the mobile user whose data had errors.
   - **Synch ID**: The ID of the synchronization session.
   - **Last Apply Date**: The date and time at which the user last synchronized the data.
   - **Error Count**: The number of errors that have occurred.
   - **Times Applied:** This column displays the number of times the transaction was reapplied unsuccessfully, after it was deferred. For a new upload error, the value is "1."
   - **Select:** In this column, you can select which errors you want to purge.
4. To purge discarded errors, confirm that the errors you want to purge are checked off in the table. Alternatively, if you want to purge all errors, you can click the **Select All** button. Then click **Purge**.

Caution: Purging will permanently delete the discarded errors from the system.

Screenshot

An image of the **Discarded Upload Errors** page is provided below:



6.7 Managing Passed Upload Errors

Passed upload errors are transactions have been successfully reapplied. Please note that only transactions that originally failed and later were successfully reapplied are

shown in the Upload Errors tab. Use this procedure to manage passed upload errors.

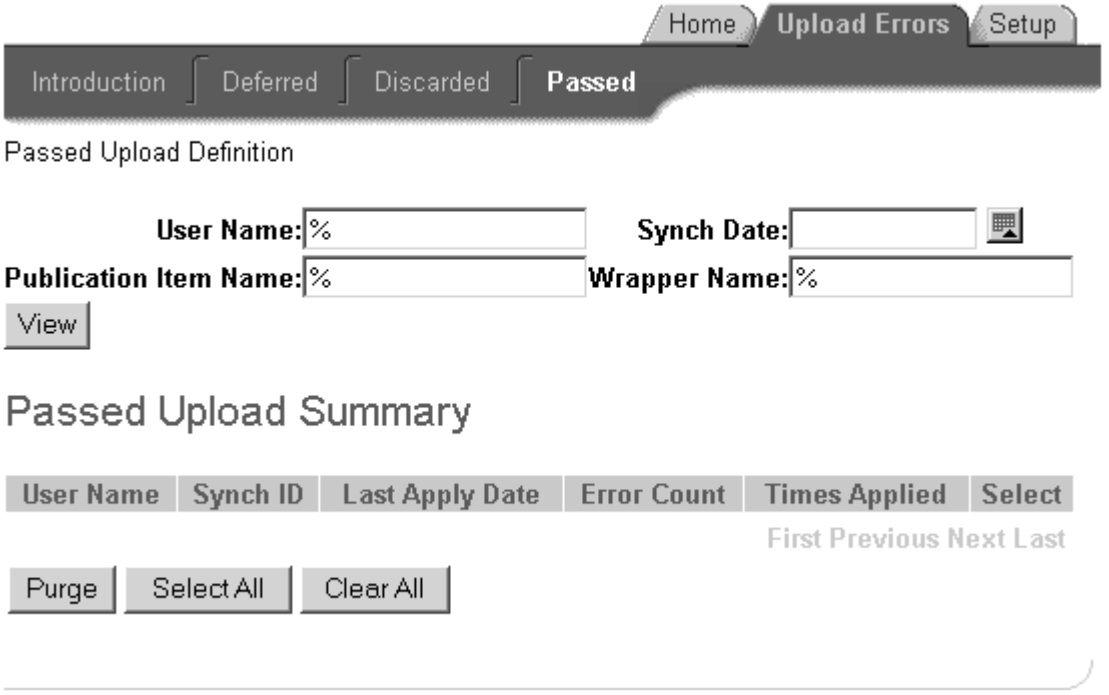
Steps

1. In the Mobile Application Foundation Administration Console, choose the **Upload Errors** tab and then the **Passed** subtab to navigate to the **Passed Upload Errors** page.
2. If desired, you can search for errors by one or more of the provided parameters: user name, synchronization date, publication name, publication item name, or the name of the wrapper that failed. The default result set displays all passed errors. To perform a search, enter values into the appropriate fields, then click **View**. You can use the percent sign (%) as a wildcard character.
3. Review the **Passed Upload Errors Summary** table, which lists passed user synchronization session errors. The table contains the following columns:
   - **User Name**: The name of the mobile user whose data had errors.
   - **Synch ID**: The ID of the synchronization session.
   - **Last Apply Date**: The date and time at which the user last synchronized the data.
   - **Error Count**: The number of errors that have occurred.
   - **Times Applied:** This column displays the number of times the transaction was re-applied unsuccessfully, after it was deferred. For a new upload error, the value is "1."
   - **Select:** In this column, you can select which errors you want to purge.
4. To purge passed errors, confirm that the errors you want to purge are checked off in the table. Alternatively, if you want to purge all errors, you can click the **Select All** button. Then click **Purge**.

> **Caution:** Purging will permanently delete the discarded errors from the system.

Screenshot

An image of the **Passed Upload Errors** page is provided below:



6.8 Using the Oracle9i Lite Administration Screens

Oracle9iLite provides an administrative console. The console can be accessed at:

```
http://<mobile_server_IP_address>/webtogo
```

The administrator can log in to the Mobile Server console. The console can be used for the following tasks:

- Changing password for the Mobile Server Repository
- Stopping or starting the Mobile Server, independent of the Apache listener
- Enabling or disabling trace options

Please refer to the *Oracle9i Lite Mobile Server Administration Guide* for details.

Note: Whenever you change the Mobile Server Repository password using this administration screen, you should also make sure to set the profile option ASG_CONS_PASSWD with the new password. Failure to do so will cause errors while applying Mobile Application patches.

A

Mobile Migration

This appendix is intended for existing users of Oracle Mobile Field Service/Laptop and Oracle Mobile Field Service/Palm. It explains the procedures that are required to upgrade from 11.5.7 and earlier releases of Oracle Mobile Field Service/Laptop and Oracle Mobile Field Service/Palm to release 11.5.9.

Attention: If you are new to mobile applications or are upgrading from 11.5.8 to 11.5.9, then you can ignore this appendix.

Note: Oracle Field Sales/Laptop (ASL) and Oracle Field Sales/Palm (ASP) are no longer available as of Oracle Applications release 11.5.8. If you use these products, you must migrate to Sales Offline Edition. Refer to the Oracle Sales Offline Edition User Guide on Oracle MetaLink.

A.1 Background

Release 11.5.9 provides the Oracle Mobile Application Foundation, which replaces the Oracle CRM Gateway for Mobile Devices. The Oracle CRM Gateway for Mobile Devices was used in releases prior to 11.5.8. As of 11.5.8 and later, the Oracle Mobile Field Service/Laptop and Oracle Mobile Field Service/Palm applications use the new Oracle Mobile Application Foundation.

The Oracle Mobile Application Foundation is not compatible with pre-11.5.8 releases of Oracle Mobile Field Service/Laptop or Oracle Mobile Field Service/Palm. Also, the 11.5.8 and 11.5.9 releases of Oracle Mobile Field Service/Laptop or Oracle Mobile Field Service/Palm are not compatible with the pre-11.5.8 Oracle CRM Gateway for Mobile Devices infrastructure. You must upgrade the applications and the foundation at the same time.

When upgrading from the 11.5.6 or 11.5.7 releases of Oracle Mobile Field Service/Laptop, there are only minor changes to the Oracle Mobile Field Service/Laptop application. When upgrading from the 11.5.4 or 11.5.5 releases of Oracle Mobile Field Service/Laptop, the Visual Basic laptop application is replaced by the HTML laptop application.

The migration steps are the same in all cases.

A.2 Technology and Requirements

A.2.1 New Mobile Architecture

Please refer to Chapter 1 of this guide for more information on the new architecture of the Oracle Mobile Application Foundation.

A.3 Supported System Configurations

Make sure that your upgraded enterprise system will conform to the following parameters before starting the migration.

- Database: Versions 8.1.7 or higher.
- JDK 1.3 on the HTTP Server and concurrent processing nodes.

A.4 Migration Overview

The new mobile architecture is fundamentally different from the old one. As a result, the migration consists of a clean removal of the old infrastructure and applications and a fresh install of the new infrastructure and applications. ***However, all data for users will remain intact.***

The administrator will need to follow these steps to successfully migrate users to the new mobile solution.

- Synchronizing All Mobile Users

- Removing the Old Mobile Applications and Database
- Applying the 11.5.9 Patches
- Running the Migration Script
- Implementing the Mobile Application Foundation
- Implementing Mobile Field Service/Laptop and Oracle Mobile Field Service/Palm
- Assigning New Mobile Responsibilities
- Downloading and Installing the New Mobile Applications

A.5 Migration Tasks

A.5.1 Step 1: Synchronizing All Mobile Users

All users need to synchronize to submit their outstanding transactions into the system. Once all transactions have been synchronized and resolved, users must refrain from using the application until the migration is complete.

At this time, you should review any deferred upload errors and reapply or discard them. Any pending upload errors that are not resolved now will be lost.

A.5.2 Step 2: Removing the Old Mobile Applications and Database

All users need to remove the old laptop or Palm application files and delete their local Oracle8i Lite or Oracle9i Lite databases.

Laptop

Laptop users need to follow these steps.

1. Click the **Start** button from the taskbar.
2. Select **Programs**.
3. Select **Oracle 9iLite > Uninstall Webtogo.**

A pop-up box will appear, prompting you to remove or keep the application odb files. Please select **Yes** to remove the application database files.

Palm

Palm users need to delete the following applications:

- mSync
- Oracle Field Service Palm

Follow these steps to delete the Oracle Mobile Field Service/Palm application without doing a hard reset:

1. On the Palm device, select **Menu**, then select **Delete**. If any of the following files are present, then delete them:
   - apps
   - ocSysDB_<mSync_username>
   - mSql
   - mSync
   - ocCompressTransport
   - ocHttpTransport
   - odbc10
   - okapi10
   - okSysDB
   - okTransLog
   - OL_LOG
   - OL_PREFS
   - OlLibC10
   - OlLibCHeap
   - FieldService
2. Do a soft reset of the Palm device.

A.5.3 Step 3: Applying the 11.5.9 Patches

For patch information, please refer to Oracle*MetaLink* Note 226830.1.

A.5.4 Step 4: Running the Migration Script

In this step you run the migration script to remove definitions of mobile users, the CRM Gateway for Mobile Devices, snapshot logs, and AQ objects from the Applications database. These objects are not required after the migration to release

11.5.9. This script also automatically migrates the mobile users in the CRM Gateway for Mobile Devices to release 11.5.9 by assigning the new responsibilities. To run the migration script, do the following procedure:

Steps

1. Connect to the enterprise database.
2. Log in as user **apps.**
3. Run the script `$ASG_TOP/patch/115/sql/asgmigrt.sql`.
4. Spool the output of this script to review the log.

A.5.5 Step 5: Installing and Configuring Oracle9i Lite

Use Section 3.2, "Installing and Configuring Oracle9i Lite."

A.5.6 Step 6: Implementing the Oracle Mobile Application Foundation

Use Chapter 4, "Implementation Tasks."

A.5.7 Step 7: Implementing Mobile Field Service/Laptop and Oracle Mobile Field Service/Palm

If you are using the Oracle Mobile Field Service/Laptop application, follow the instructions in the *Oracle Mobile Field Service Implementation Guide for Laptops*.

If you are using the Oracle Mobile Field Service/Palm application, follow the instructions in the *Oracle Mobile Field Service Implementation Guide for Palm Handhelds*.

A.5.8 Step 8: Assigning New Mobile Responsibilities

With the new Oracle Mobile Application Foundation, users are no longer managed separately. Access to the mobile applications is based on the users being granted the appropriate mobile responsibility. Mobile user names and passwords are the same as for the online applications.

The script in Step 4 migrates the mobile users. However, you should inform the users of their user names and passwords.

A.5.9 Step 9: Downloading and Installing the New Mobile Applications

Follow the instructions in Appendix A, "Installing the Laptop Application" of the *Oracle Mobile Field Service Implementation Guide for Laptops* and Appendix A, "Installing the Client Application" of the *Oracle Mobile Field Service Implementation Guide for Palm Handhelds* to set up the application and do a fresh synchronization for the client laptop and Palm applications.

Finally, test the application(s).