Oracle® Application Server Wireless

Administrator's Guide

10*g* Release 2 (10.1.2)

B13820-02

August 2005

Oracle Application Server Wireless Administrator’s Guide, 10*g* Release 2 (10.1.2)

B13820-02

Contents

4 Managing Oracle Sensor Edge Services

5 Managing Users

6 Managing Content

7 Administering Mobile Studio

Part III Configuration and Integration

9 OracleAS Wireless Gateway Configuration

10 OracleAS Wireless Security

11 Mobile Single Sign-On

12 Activity Logging

13 Optimizing Oracle Application Server Wireless

14 Load Balancing and Failover

15 Globalization

16 Integrating OracleAS Wireless with Other Components

A Troubleshooting Oracle Application Server Wireless

Glossary

Index

Preface

The Administrator’s Guide discusses how you can use Oracle Application Server Wireless to develop and deliver mobile applications.

Audience

This book is intended for OracleAS Wireless administrators.

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.

Related Documents

You can also find information on Oracle Application Server Wireless through these resources:

- Oracle Technology Network—Oracle Technology Network is dedicated to providing developers the best information on Oracle's products and technologies. Visit:

  http://www.oracle.com/technology/tech/wireless/

- *Oracle Application Server Wireless API Reference*
- *Oracle Application Server Wireless Developer's Guide*
- Support—Visit: http://www.oracle.com/support/

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introducing Oracle Application Server Wireless

This chapter, through the following sections describes Oracle Application Server Wireless and this user guide:

- Section 1.1, "Overview of OracleAS Wireless Tools"
- Section 1.2, "Using This Guide"

1.1 Overview of OracleAS Wireless Tools

OracleAS Wireless provides a complete set of Web-based tools, which provide functions for developing and publishing mobile applications, creating mobile users, providing help desk support, and managing the OracleAS Wireless server. These tools each include step-by-step wizards, which enable you to quickly accomplish any task. The wizard pages include in-line hints and tips which provide information for new users to quickly learn the tools. The online help enables experienced users to utilize the advanced features.

Out of the box, OracleAS Wireless provides the following tools:

- System Manager

  The System Manager tool enables users to configure, manage and monitor the performance of OracleAS Wireless.

- Service Manager

  The Service Manager enables users to create different types of OracleAS Wireless applications, such as HTTP-based applications or J2ME MIDlets.

- Content Manager

  The Content Manager enables users to publish and customize Service Manager-created applications to OracleAS Wireless users.

- Sensor Services Tool

  The Sensor Services tool enables users to create and manage the objects of the Oracle Sensor Edge Server.

- Foundation Manager

  The Foundation Manager enables users to create and modify the such objects as devices, transformers, adapters, regions, digital rights policies, and API scan policies

- User Manager

The User Manager provides user-support tasks, such as creating a new user, resetting the PIN and password for a user or viewing user statistics.

- OracleAS Wireless Customization Portal

  The OracleAS Wireless Customization Portal enables end-users to personalize OracleAS Wireless applications and manage their personal profiles, including their devices and Location Marks. The Customization Portal can be used as the out-of-box Web-based portal along with the device-based portal.

The OracleAS Wireless Tools are role-specific; OracleAS Wireless users can only access the tool which corresponds to the role or roles that they have been granted. These user roles, which are described in Table 1–1, span all of the OracleAS Wireless resources, from server management, application development, application publishing, and help desk to subscription to the OracleAS Wireless applications. Because these tools are Web-based, they require no client-side installation. After installing and starting the OracleAS Wireless server, multiple users can access the OracleAS Wireless tools through normal desktop browsers.

You do not need to manually configure any server files or code with APIs to access the out-of-box features of the OracleAS Wireless server, unless you want to expand the OracleAS Wireless server functions.

Table 1–1 OracleAS Wireless User Roles

User Role	Description	Available Tools
Application Developer	Users assigned the Application Developer role perform the following functions: ■ Create, modify, delete and test applications. ■ Publish applications to the Application Developer's folder. ■ Create, modify, and delete notifications. ■ Create, modify, and delete data feeders. ■ Register and delete J2ME Web services. ■ Develop preset definitions.	Service Manager
Foundation Developer	Users assigned the Foundation Developer role perform the following functions: ■ Create, modify, and delete devices. ■ Create, modify, and delete transformers. ■ Create, modify, and delete regions. ■ Create, modify, and delete digital rights policies. ■ Create, modify, and delete API scan policies.	Foundation Manager
Content Manager	Users assigned the Content Manager role perform the following functions: ■ Manage application folders and bookmarks. ■ Create application links based on Application Developer-created applications. ■ Create notifications based on alerts (now deprecated). ■ Create application categories and associate access points with them. ■ Create a user-home folder rendering scheme, such as setting the sorting order for applications.	Content Manager

Table 1–1 (Cont.) OracleAS Wireless User Roles

User Role	Description	Available Tools
System Administrator	Users assigned the System Administrator role perform configuration management and performance monitoring for various OracleAS Wireless servers. The OracleAS Wireless servers are deployed both as OC4J (OracleAS Containers for J2EE) applications and as standalone Java applications.	System Manager. This tool is packaged with Oracle Enterprise Manager and is accessed through the Application Server Control.
User Manager	Users assigned the User Manager role perform the following functions: ■ Manage users by providing such Help Desk functions as editing a user profile, resetting passwords and PINs, and creating or deleting users. ■ Manage user access privileges. ■ View application links assigned to users. ■ Manage user devices. ■ Search for users. ■ View overview information of users.	User Manager
Sensor Services Administrator	User assigned the Sensor Services Administrator role manage the devices and filters used with the Oracle Sensor Edge Server. These functions include: ■ Create drivers for Oracle Sensor Edge Server devices ■ Create filters used with Oracle Sensor Edge Server devices ■ Manage the Oracle Sensor Edge Server devices assigned to the Edge Server processes	Sensor Services Tool
End User	Users assigned the end user role are the consumers of OracleAS Wireless services. End-users create their own accounts when they register with OracleAS Wireless using the OracleAS Wireless Customization. End users can also customize their own applications either from a desktop or from a device. Customization for end-users includes: ■ Customize applications, download J2ME applications, subscribe to notifications. ■ Manage devices. ■ Manage location marks and location settings. ■ Manage contact rules. Mobile studio users also have the end user role; a user belonging to the *StudioUser* group can access the Mobile Studio. Every OracleAS Wireless user is granted the Mobile Customer Role by default. This role is implicit to all users.	Wireless Customization Portal Mobile Studio (for users assigned to the *StudioUser* group)

1.2 Using This Guide

In addition to the OracleAS Wireless Tools (described in Section 1.1), this guide describes the following:

- OracleAS Wireless server and site configuration.

- OracleAS Wireless gateway configuration for voice and messaging communications.
- Security support for OracleAS Wireless.
- Mobile user Single Sign-On (SSO) support.
- Activity logging for the Multi-Channel Server, Notification Engine, Messaging Server, data feeder engine, and the Async Listener.
- The tuning knobs for enhancing the performance of OracleAS Wireless, such as those for JVM as well as the Oracle Application Server high-availability components, Oracle HTTP Server (OHS), OracleAS Containers for J2EE (OC4J) OracleAS Wireless Instance, and Oracle Process Management and Notification (OPMN).
- Load balancing and failover enabled by clustering and high availability.
- Multi-locale and multi-encoding support.
- Integration with other components.

Part I

Overview

This section includes the following chapters:

- Chapter 1, "Introducing Oracle Application Server Wireless"
- Chapter 2, "Verifying the OracleAS Wireless Installation"

2

Verifying the OracleAS Wireless Installation

This chapter describes how to access the OracleAS Wireless tools and the Device Customization portal to verify the proper installment and functioning of these applications. See the *Oracle Application Server Wireless Developer's Guide* for information on installing OracleAS Wireless.

This chapter includes the following sections:

- Section 2.1, "Starting the Oracle Application Server Wireless Server"
- Section 2.2, "Accessing the OracleAS Wireless Customization Portal"
- Section 2.3, "Accessing the OracleAS Wireless Tools"
- Section 2.4, "Re-Registering the OracleAS Wireless Portal Services URL Reference in OracleAS Portal"

2.1 Starting the Oracle Application Server Wireless Server

Before OracleAS Wireless users can access the OracleAS Wireless development tools, the administrator must start the server using the OracleAS Wireless system management page (accessed through the Oracle Enterprise Manager Application Server Control).

To log into the Application Server Control and access the management functions for OracleAS Wireless:

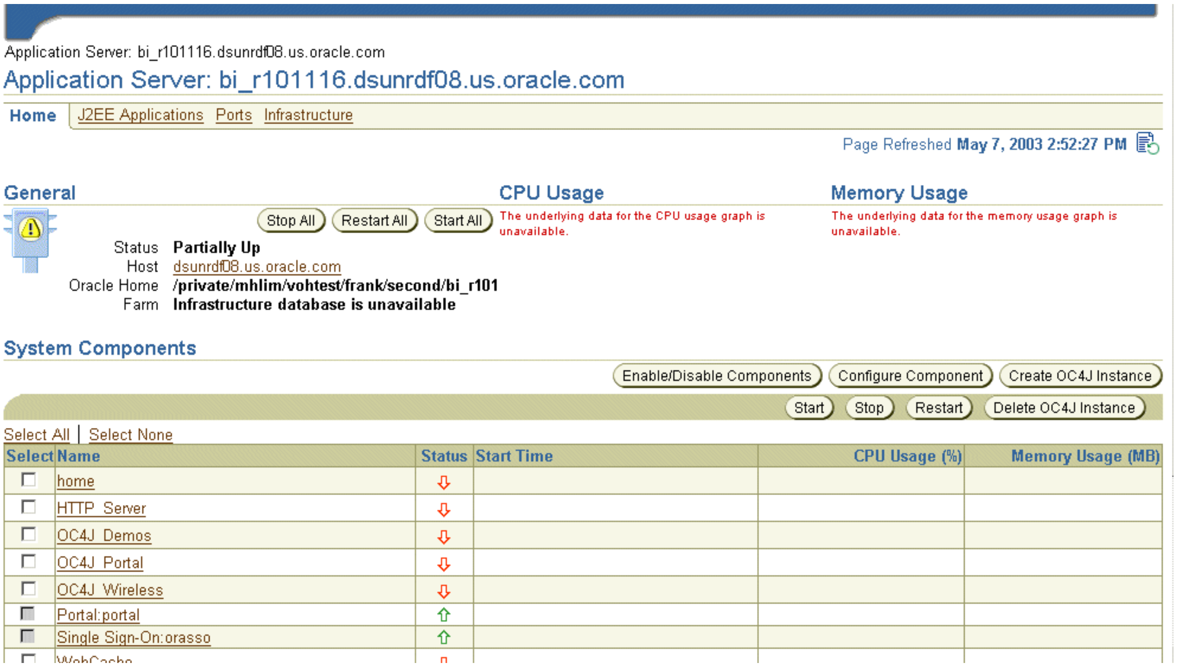
1. Enter the following URL into a browser:

```
http://Server:1810
```

> **Note:** The default ports are 1810 and 1811. The port number range is 1812 to 1820. To ensure that you are using the correct port number, check the port number for Oracle Application Server Wireless stored in `[Oracle home]/install/portlist.ini`.
>
> Refer to the *Oracle9i Application Server Administrator's Guide* for more information on port usage.

2. Enter the administrator user name and password. The Oracle Enterprise Manager *Home* page appears (Figure 2–1).

Figure 2–1 The OracleAS Enterprise Manager Home Page (Partial View)



Application Server: bi_r101116.dsunrdf08.us.oracle.com

Application Server: bi_r101116.dsunrdf08.us.oracle.com

Page Refreshed **May 7, 2003 2:52:27 PM**

General

Status **Partially Up**
Host dsunrdf08.us.oracle.com
Oracle Home **/private/mhlim/vohtest/frank/second/bi_r101**
Farm **Infrastructure database is unavailable**

CPU Usage

The underlying data for the CPU usage graph is unavailable.

Memory Usage

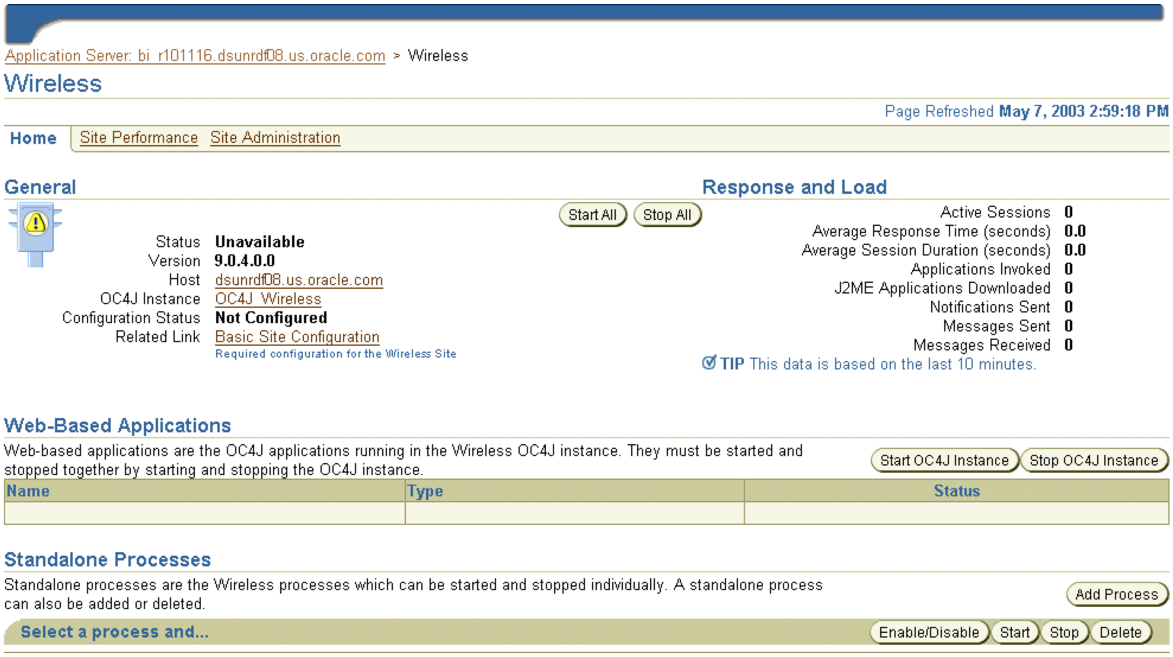
The underlying data for the memory usage graph is unavailable.

System Components

Select	Name	Status	Start Time	CPU Usage (%)	Memory Usage (MB)
□	home	⇩			
□	HTTP_Server	⇩			
□	OC4J_Demos	⇩			
□	OC4J_Portal	⇩			
□	OC4J_Wireless	⇩			
□	Portal:portal	⇧			
□	Single Sign-On:orasso	⇧			
□	WebCache	⇩			

3. From the *System Components* table, select *Wireless*. The *Home* page of the OracleAS Wireless tool, System Manager, appears (Figure 2–2).

Figure 2–2 The System Manager Home Page (Partial View)



Application Server: bi_r101116.dsunrdf08.us.oracle.com > Wireless

Wireless

Page Refreshed **May 7, 2003 2:59:18 PM**

General

Status **Unavailable**
Version **9.0.4.0.0**
Host dsunrdf08.us.oracle.com
OC4J Instance OC4J_Wireless
Configuration Status **Not Configured**
Related Link Basic Site Configuration
Required configuration for the Wireless Site

Response and Load

Active Sessions	**0**
Average Response Time (seconds)	**0.0**
Average Session Duration (seconds)	**0.0**
Applications Invoked	**0**
J2ME Applications Downloaded	**0**
Notifications Sent	**0**
Messages Sent	**0**
Messages Received	**0**

TIP This data is based on the last 10 minutes.

Web-Based Applications

Web-based applications are the OC4J applications running in the Wireless OC4J instance. They must be started and stopped together by starting and stopping the OC4J instance.

Name	Type	Status

Standalone Processes

Standalone processes are the Wireless processes which can be started and stopped individually. A standalone process can also be added or deleted.

2.1.1 Configuring the OracleAS Wireless Server

If the *Configuration Status* displays as *Not Configured* (as it would, for example, during the initial session after the OracleAS Wireless repository has been installed), an administrator can configure the OracleAS Wireless Server within minutes using a two-step configuration wizard accessed through the Basic Site Configuration link. The administrator need only add the name and port values for the HTTP and HTTPS proxy server, the address for the OracleAS Wireless access points for the Async Listener, SMS and IM servers, and then set the correct time zone for the OracleAS Wireless Server. For more information, see Section 3.3 in Chapter 3, "Managing the OracleAS Wireless Server".

Note: If the *Configuration Status* indicates that the OracleAS Wireless sever has not been configured, then the administrator must configure the server before starting the processes.

2.2 Accessing the OracleAS Wireless Customization Portal

This section describes how to log into the OracleAS Wireless Customization Portal.

Before using the OracleAS Wireless Customization Portal, you must access the login page by entering the following URL in a browser:

```
http://<host>:<port>/mobile/Login.uix
```

For example, you access the login page through the following URL:

```
http://hostname:7777/mobile/Login.uix
```

Note: 7777 is the default port number for Oracle Application Server Wireless. The port number range is 7777 to 7877. To ensure that you are using the correct port number, check the port number for Oracle Application Server Wireless stored in `[Oracle home]/install/portlist.ini`. For more information on port usage, see the *Oracle9i Application Server Administrator's Guide*

After you enter the URL, the login page for the Wireless Customization Portal appears. This page includes the described in Table 2–1.

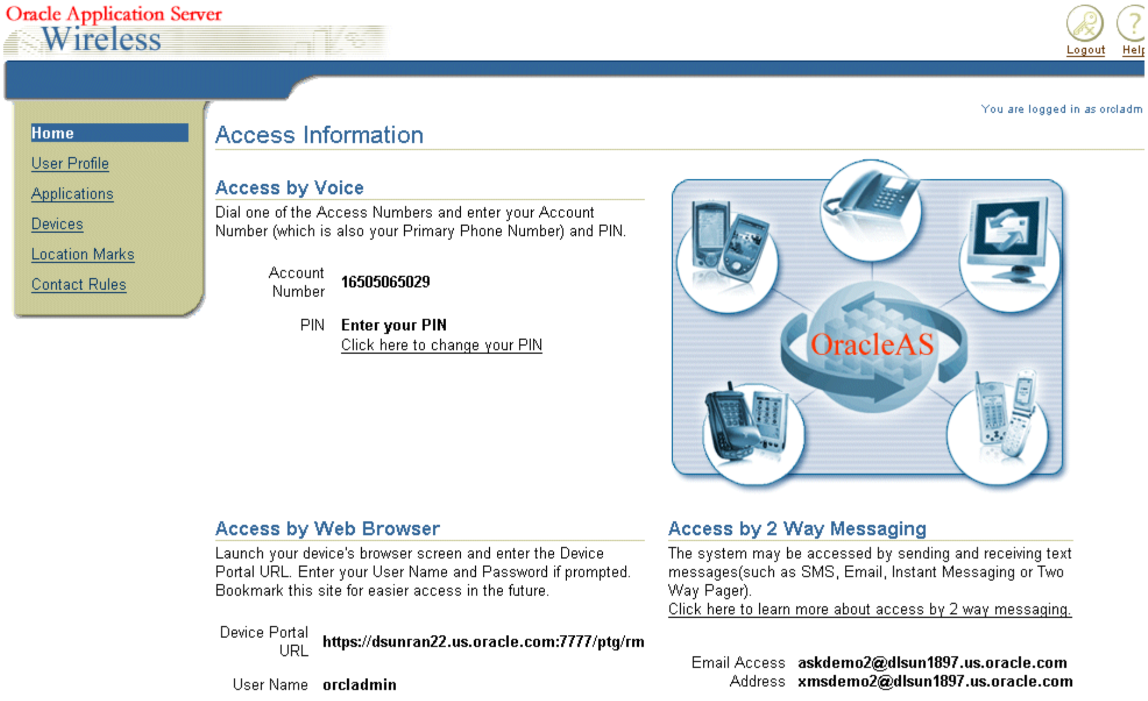
Table 2–1 Login Screen Buttons

Button	Description
Login	Clicking this button logs you in after you have entered the correct user name and password.
Help	Clicking this button displays a list of help topics.
Page Help	Clicking this button displays help topics specific to this screen.

1. Enter your user name and then enter your password. If you are an administrator, enter *orcladmin* as your user name. (The password is set during installation, but can be changed with the User Manager.)
2. Click **Login**.

After you successfully log in, the Welcome screen appears (Figure 2–3), which includes your addresses for accessing Oracle Application Server Wireless applications.

Figure 2–3 The Welcome Screen of Wireless Customization (Partial View)



You are logged in as orcladm

Access Information

Access by Voice

Dial one of the Access Numbers and enter your Account Number (which is also your Primary Phone Number) and PIN.

Account Number 16505065029

PIN **Enter your PIN**
Click here to change your PIN

Access by Web Browser

Launch your device's browser screen and enter the Device Portal URL. Enter your User Name and Password if prompted. Bookmark this site for easier access in the future.

Device Portal URL **https://dsunran22.us.oracle.com:7777/ptg/rm**

User Name **orcladmin**

Access by 2 Way Messaging

The system may be accessed by sending and receiving text messages(such as SMS, Email, Instant Messaging or Two Way Pager).
Click here to learn more about access by 2 way messaging.

Email Access Address **askdemo2@dlsun1897.us.oracle.com**
xmsdemo2@dlsun1897.us.oracle.com

2.3 Accessing the OracleAS Wireless Tools

This section describes how to log into the Oracle Application Server Wireless Tools to access the User Manager, Service Designer, Foundation Manager, and Content Manager. If you access the Oracle Application Server Wireless Tools in standalone mode, then you can also access the System Manager (described in Section 2.1).

Access the login page for the Oracle Application Server Wireless Tools through the following URL:

```
http://<host>:<port>/webtool/login.uix
```

For example, you access the login page through the following URL:

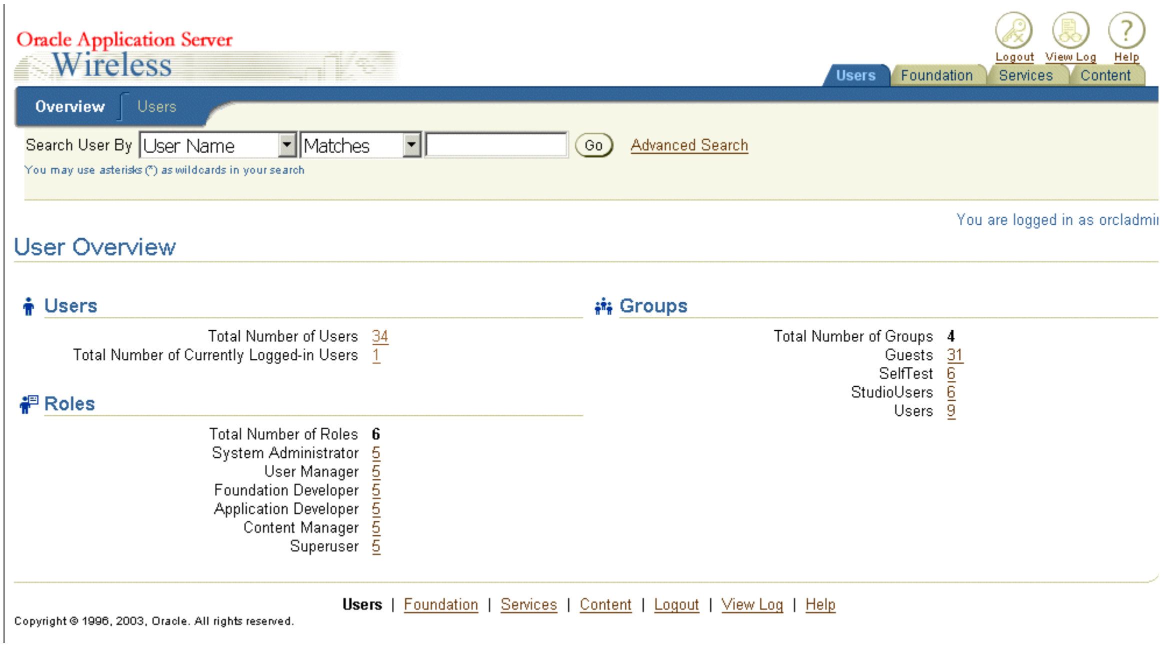
```
http://hostname:7777/webtool/login.uix
```

Note: 7777 is the default port number for Oracle Application Server Wireless. The port number range is 7777 to 7877. To ensure that you are using the correct port number, check the port number for Oracle Application Server Wireless stored in [Oracle home]/install/portlist.ini. For more information on port usage, see the Oracle Application Server Administrator's Guide

---

Enter your user name and then enter your password. If you are an administrator, enter *orcladmin* as your user name. (The password is set during installation, but can be changed with the User Manager.) The Oracle Application Server Wireless Tools appear, with the User Manager displaying by default (as displayed in Figure 2–4).

Figure 2–4 The Oracle Application Server Wireless Tools (with User Manager Displayed)



Search User By User Name ▾ Matches ▾ Go

You may use asterisks (*) as wildcards in your search

You are logged in as orcladmi

User Overview

Users

Total Number of Users	34
Total Number of Currently Logged-in Users	1

Roles

Total Number of Roles	6
System Administrator	5
User Manager	5
Foundation Developer	5
Application Developer	5
Content Manager	5
Superuser	5

Groups

Total Number of Groups	4
Guests	31
SelfTest	6
StudioUsers	6
Users	9

Users | Foundation | Services | Content | Logout | View Log | Help

Copyright © 1996, 2003, Oracle. All rights reserved.

2.4 Re-Registering the OracleAS Wireless Portal Services URL Reference in OracleAS Portal

Installing Oracle Application Server Wireless 10*g* Release 2 (10.1.2.02) against Oracle Application Server Wireless 9.0.2 re-registers the OracleAS Wireless portal service URL and might prevent OracleAS Portal from responding to mobile requests. To avoid this problem, you must re-register the URL in the Oracle Application Server Portal as follows:

1. From the Portal Builder, click the **Administer** tab.
2. In the *Services* portlet, click **Global Settings**.
3. Click the **Mobile** tab.
4. Enter the OracleAS Wireless URL in the *OracleAS Wireless 10g Wireless Portal Service URL* field.
5. Click **OK**.

For more information about configuring mobile settings and updating the Oracle Application Server Wireless service URL in OracleAS Portal, refer to the *Oracle Application Server Portal Configuration Guide.*

Part II

System Administration

This section includes the following chapters:

3

Managing the OracleAS Wireless Server

This chapter includes the following sections:

3.1 Overview of OracleAS Wireless System Management

System Administrators use the System Manager to centrally manage and configure OracleAS Wireless.

Site- and Server-Level Configuration, Management, and Performance Monitoring

The System Manager, which is part of Oracle Enterprise Manager, enables you to manage and configure OracleAS Wireless at both the site and server level. At the site level, you create a common configuration for all of the OracleAS Wireless servers. (All of the configuration data is stored in the database). At the sever level, you manage and configure the OracleAS Wireless server’s processes and monitor the server’s performance.

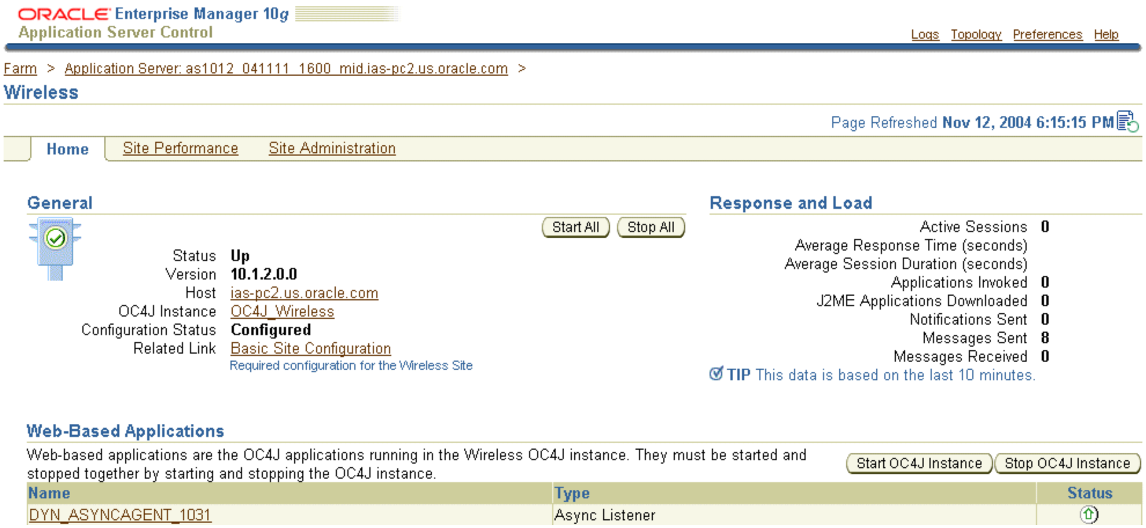
3.1.1 Using the System Manager

The System Manager is divided into three pages: the *Home* page, the *Site Performance* page, and the *Site Administration* page.

3.1.1.1 The Home Page

The *Home* page provides a the overall view of the OracleAS Wireless server. The status, processes, performance data and system logging reflect the current middle tier of the OracleAS Wireless server. The one function that is not specific to the current middle-tier instance is **Basic Site Configuration** (described in Section 3.3), which enables you to perform the minimal configuration required for the OracleAS Wireless site after OracleAS Wireless is first installed.

Figure 3–1 The Home Page of System Manager (Partial View)


ORACLE Enterprise Manager 10g
Application Server Control
Logs Topology Preferences Help
Farm > Application Server: as1012_041111_1600_mid.ias-pc2.us.oracle.com >
Wireless
Home
Site Performance
Site Administration
Start All
Stop All
Start OC4J Instance
Stop OC4J Instance

Page Refreshed Nov 12, 2004 6:15:15 PM

General

Status	Up
Version	10.1.2.0.0
Host	ias-pc2.us.oracle.com
OC4J Instance	OC4J_Wireless
Configuration Status	Configured
Related Link	Basic Site Configuration Required configuration for the Wireless Site

Response and Load

Active Sessions	0
Average Response Time (seconds)	
Average Session Duration (seconds)	
Applications Invoked	0
J2ME Applications Downloaded	0
Notifications Sent	0
Messages Sent	8
Messages Received	0

TIP This data is based on the last 10 minutes.

Web-Based Applications

Web-based applications are the OC4J applications running in the Wireless OC4J instance. They must be started and stopped together by starting and stopping the OC4J instance.

Name	Type	Status
DYN_ASYNCAGENT_1031	Async Listener	⇧

The page's *General* section displays current status of the OracleAS Wireless Server (*Up*, *Down*, or *Unavailable*), the name of the current host, the version number of OracleAS Wireless, and the configuration status of the site. The timestamp on the *Home* page reflects the current status of the data displayed on the page. You can refresh (reload) the *Home* page by clicking the *Refresh* icon.

The *Home* page is divided into the following sections:

- General
- Response and Load
- Web-Based Applications
- Standalone Processes
- Instance Configuration

General

The *General* section (illustrated in Figure 3–1) lists the current running status of the OracleAS Wireless Server, the name of the current host, and whether the server has been configured. (In Figure 3–1, *Configuration Status* is noted as *Configured*.) For more information, see Section 3.3.

Response and Load

The *Response and Load* section (illustrated in Figure 3–1) displays the following OracleAS Wireless runtime instance statistics for the last ten minutes.

- Active Sessions

  A view of the overall load in the system, showing the total number of active sessions.

- Average Response Time (seconds)

The overall service performance for the system, showing the average response time.

- Average Session Duration (seconds)

  The average duration of each session for the system. The duration of each session is computed using the login time and the expiry time.

- Applications Invoked

  The number of applications invoked.

- J2ME Applications Downloaded

  The total number of J2ME Applications (MIDlets) downloaded from OracleAS Wireless Wireless repository.

- Notifications Sent

  The number of notifications sent from the OracleAS Wireless Server.

- Messages Sent

  The total number of messages sent successfully.

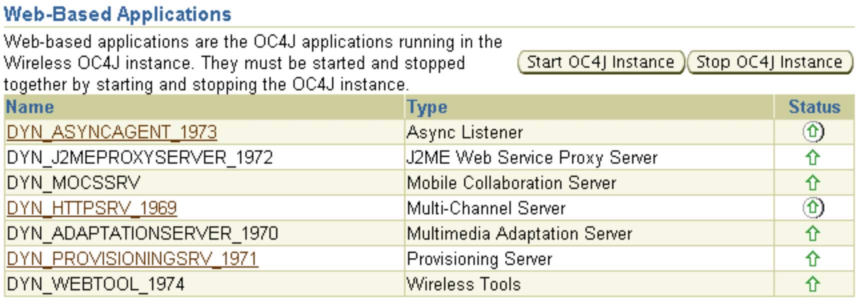
- Messages Received

  The total number of times that the transport drivers called the `onMessage` call-back method.

Web-Based Applications

This section itemizes the OC4J (OracleAS Containers for J2EE) applications in the Wireless OC4J instance in a table (illustrated in Figure 3–2), which lists them by name, type and running status. These application types vary according to the OracleAS Wireless installation. By clicking one of these applications in the *Name* column, you access pages for viewing performance statistics specific to the selected application. You can refresh the performance data displayed on these paged by clicking the **Refresh** icon (see Section 3.2.1). These applications, which are started or stopped using the **Start OC4J Instance** and **Stop OC4J Instance** buttons, are started as stopped as a group and cannot be started or stopped individually. For more information on OC4J see the *Oracle Application Server Containers for J2EE User's Guide*. For more information on the Web-based applications, see Section 3.6.1.

Figure 3–2 The Web-Based Applications Section of the Home Page



Web-Based Applications

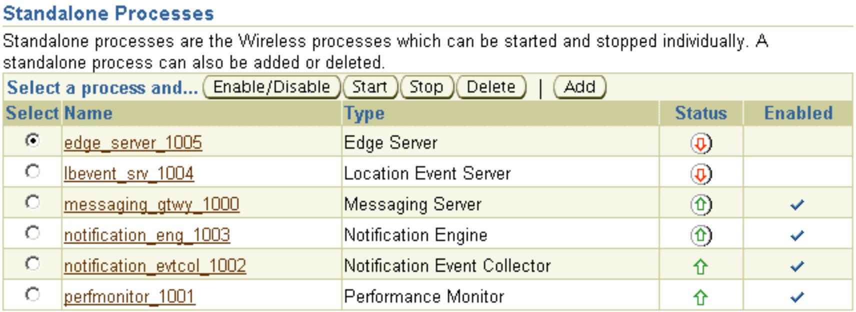
Web-based applications are the OC4J applications running in the Wireless OC4J instance. They must be started and stopped together by starting and stopping the OC4J instance.

Name	Type	Status
DYN_ASYNCAGENT_1973	Async Listener	⇧
DYN_J2MEPROXYSERVER_1972	J2ME Web Service Proxy Server	⇧
DYN_MOCSSRV	Mobile Collaboration Server	⇧
DYN_HTTPSRV_1969	Multi-Channel Server	⇧
DYN_ADAPTATIONSERVER_1970	Multimedia Adaptation Server	⇧
DYN_PROVISIONINGSRV_1971	Provisioning Server	⇧
DYN_WEBTOOL_1974	Wireless Tools	⇧

Standalone Processes

This section (illustrated in Figure 3–3) lists the OracleAS Wireless process types, which vary according to the installation.

Figure 3–3 The Standalone Process Section of the Home Page



Standalone Processes

Standalone processes are the Wireless processes which can be started and stopped individually. A standalone process can also be added or deleted.

Select a process and...

Select	Name	Type	Status	Enabled
●	edge_server_1005	Edge Server	↓	
○	lbevent_srv_1004	Location Event Server	↓	
○	messaging_gtwy_1000	Messaging Server	↑	✓
○	notification_eng_1003	Notification Engine	↑	✓
○	notification_evtcol_1002	Notification Event Collector	↑	✓
○	perfmonitor_1001	Performance Monitor	↑	✓

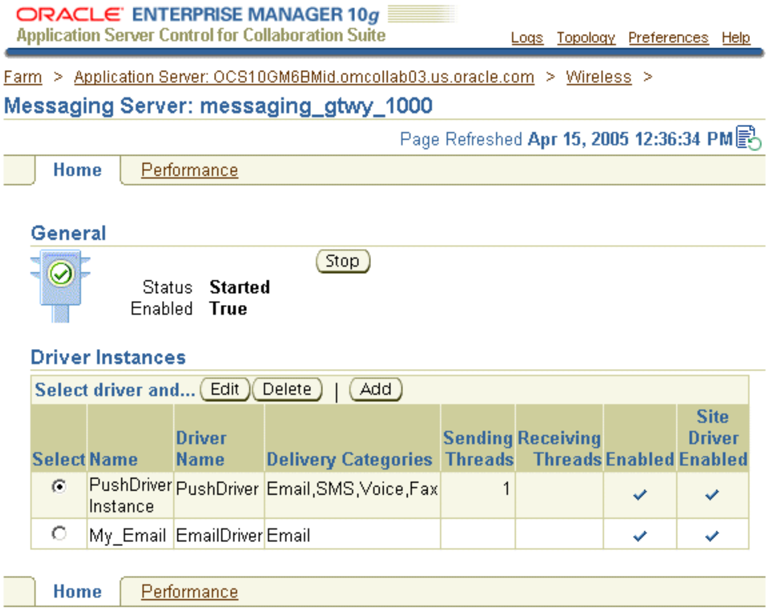
These processes are listed in a table by process name, process type (described in Section 3.6.2), running status, and if the process has been enabled. Table 3–1 describes the buttons used to manage the standalone processes.

Table 3–1 Management Functions for Standalone Processes

Function	Description
Enable/Disable	Enables or disables a selected process.
Start	Starts a selected process.
Stop	Stops the selected process.
Delete	Deletes a selected process.
Add	Adds a process.

By clicking one of these processes in the *Name* column of the table, you access detail pages (such as Figure 3–4) that enable you to start and stop the process, view its performance statistics, and configure it. You can refresh data on the detail pages by clicking the **Refresh** icon (see Section 3.2.1).

Figure 3–4 A Process Detail Page



ORACLE ENTERPRISE MANAGER 10g
Application Server Control for Collaboration Suite
Logs
Topology
Preferences
Help

Farm > Application Server: OCS10GM6BMid.omcollab03.us.oracle.com > Wireless >

Messaging Server: messaging_gtwy_1000

Page Refreshed **Apr 15, 2005 12:36:34 PM**

General

Status **Started**
Enabled **True**

Driver Instances

Select driver and...

Select	Name	Driver Name	Delivery Categories	Sending Threads	Receiving Threads	Enabled	Site Driver Enabled
●	PushDriver Instance	PushDriver	Email,SMS,Voice,Fax	1		✓	✓
○	My_Email	EmailDriver	Email			✓	✓

Instance Configuration

From this section (pictured in Figure 3–5), you can configure the logging directory, view the log file, and configure the URLs for the current OracleAS Wireless instance or middle tier. For more information on configuring the logging directory, see Section 3.4. For information on setting the instance URLS, see Section 3.5.

Figure 3–5 The Instance Configuration Section of the Home Page

Instance Configuration

Logging Directory

Instance URLs

3.1.1.2 The Site Performance Page

The *Performance* page displays the performance data of the OracleAS Wireless site. For information, see Section 3.8.

3.1.1.3 The Site Administration Page

The *Site Administration* page is divided into the *General Configuration* (Figure 3–18), *Component Configuration* (Figure 3–19), and *Utilities* sections.

General Configuration

This *General Configuration* section enables you set the common configuration the entire OracleAS Wireless site, such as the JDBC connection pool, system log level, locale, and OracleAS Wireless server URLs. For more information, see Section 3.9.

Component Configuration

The *Component Configuration* section enables you to configure specific components, such as the Multi-Channel Server, the Async Listener, the Messaging Server and Messaging Server drivers, the Notification Engine, and the Provisioning Server. For more information on configuring these components, see Section 3.10.

Utilities

The *Site Administration* page also includes a *Utilities* section, which provides functions for uploading and downloading repository objects. For more information, see Section 3.11.

3.2 Logging In to the System Manager

Users granted the *System Administrator* role access the System Manager through the Oracle Enterprise Manager Application Server Control by entering the following URL into browser:

```
http://<server>:1810
```

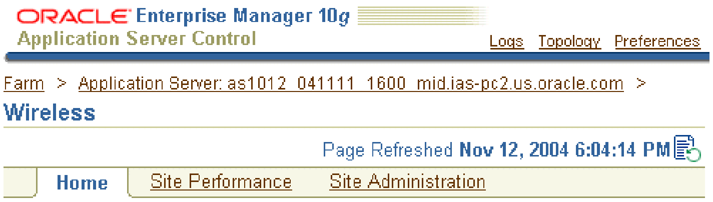
Note: The default port is 1810.

After you log into the Oracle Enterprise Manager, select the *Wireless* component from the *System Components* table. The System Manager appears and defaults to the *Home* page (Figure 3–1). For more information on the Oracle Enterprise Manager, refer to *Oracle Enterprise Manager Concepts*.

3.2.1 Refreshing the System Manager Screens

The *Home*, *Site Performance* and *Administration* pages each have a timestamp that indicates the status of the data displayed. To update this data, click the *Refresh* icon (illustrated in Figure 3–6). Refreshing the *Home* and *Site Performance* pages reloads the performance or status information, not the configuration data. To refresh the configuration data (that is, to force the configuration data to be reloaded from the database), click the *Refresh* icon on the *Site Administration* page. The timestamp on the *Home* and *Site Performance* pages displays the current time, because the data is retrieved in real-time; the timestamp on the *Site Administration* page, however, displays the last time that the configuration data was loaded from the database. To refresh the page, you must either click the **Refresh** icon or update some configuration data.

Figure 3–6 The Refresh Icon on the Home Page


ORACLE Enterprise Manager 10g
Application Server Control
Logs
Topology
Preferences
Farm > Application Server: as1012_041111_1600_mid.ias-pc2.us.oracle.com >
Wireless
Page Refreshed Nov 12, 2004 6:04:14 PM
Home
Site Performance
Site Administration

3.3 Setting the Basic Site Configuration

The *Basic Site Configuration* function in the *General* section enables you to quickly configure the entire OracleAS Wireless site by providing a minimum of information.

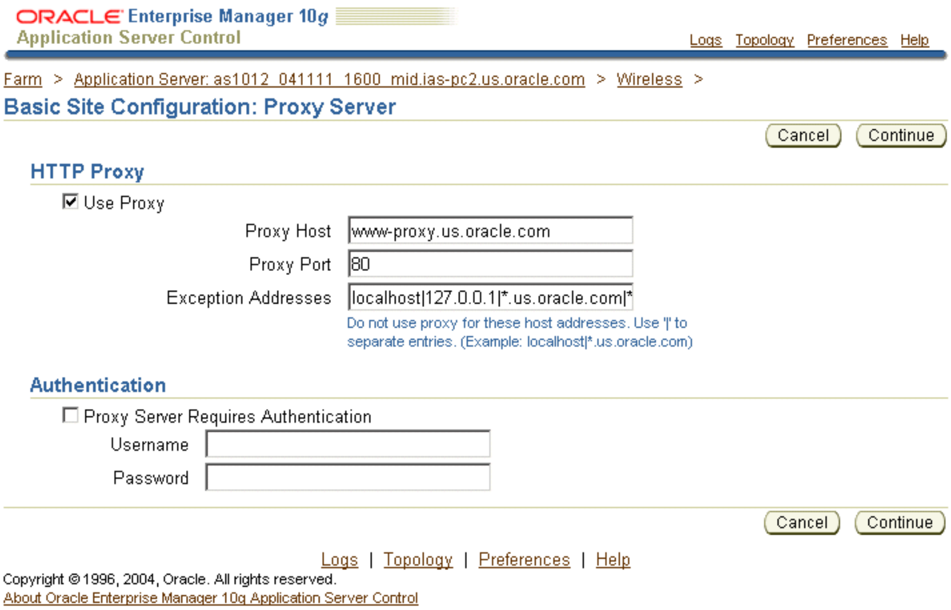
> **Note:** The OracleAS Wireless site needs only to be configured once after the installation of the first OracleAS Wireless middle tier.

Clicking **Basic Site Configuration** invokes a two-page wizard that guides you through the configuration of the OracleAS Wireless site. The pages are as follows:

- The *Proxy Server* page (Figure 3–7): On this page, you define such proxy server-related information as proxy server host name and port number. You must provide the proxy settings if the OracleAS Wireless instance resides within an intranet and must use the proxy server to direct traffic to Internet.

> **Note:** The *Proxy Server* page enables you to configure the proxy properties used by OracleAS Wireless when HTTP is required. If the OracleAS Wireless installation does not use an HTTP proxy server, then you do not have to define the parameters for this page.

Figure 3–7 Configuring the Proxy Ports for Basic Site Configuration


ORACLE Enterprise Manager 10g
Application Server Control
Logs Topology Preferences Help
Farm > Application Server: as1012_041111_1600_mid.ias-pc2.us.oracle.com > Wireless >
Cancel
Continue

Basic Site Configuration: Proxy Server

HTTP Proxy

☑ Use Proxy

Proxy Host	www-proxy.us.oracle.com
Proxy Port	80
Exception Addresses	localhost|127.0.0.1|*.us.oracle.com|*

Do not use proxy for these host addresses. Use '|' to separate entries. (Example: localhost|*.us.oracle.com)

Authentication

□ Proxy Server Requires Authentication

Username	
Password	

Cancel
Continue
Logs | Topology | Preferences | Help

Copyright © 1996, 2004, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Application Server Control

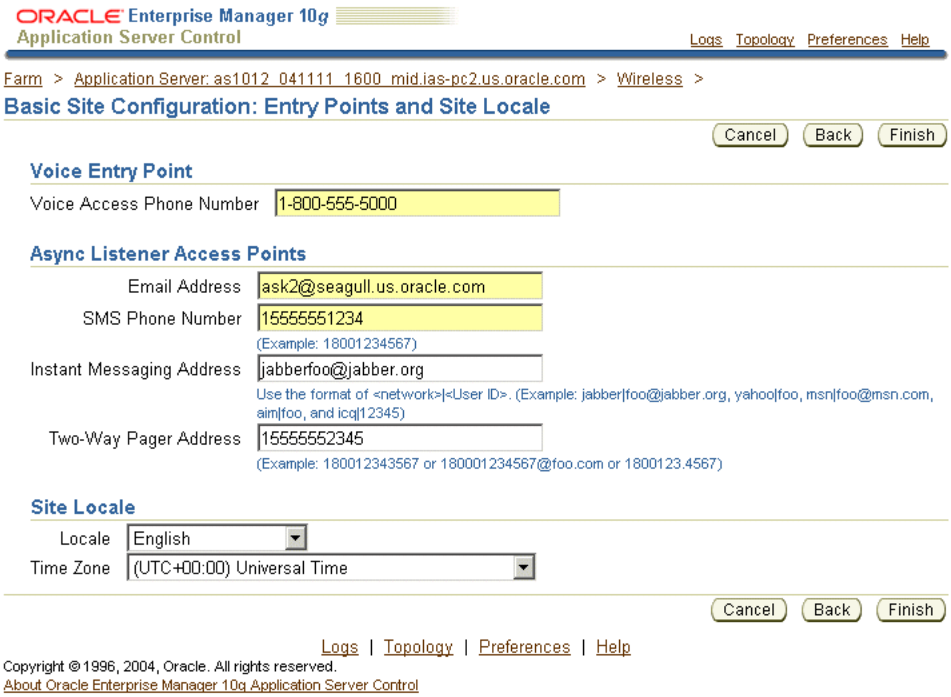
- The *Entry Points* page (Figure 3–8): You define the entry points on this page, which include the access point addresses used by different delivery channels to access the Async Listener as well as the number for the voice gateway. The values defined in this page display in the Customization Portal.

Table 3–2 The Entry Points

Parameter Name	Value
Voice Access Phone Number	The number of the voice gateway.
Email Address	The e-mail address of the Async Listener.
SMS Address	The address of the SMS server. You obtain this number from an SMS provider or aggregator. For example, enter *18005555555*.
Instant Messaging Address	The address of the instant messaging server. Enter this address as *userName@serverHost/wireless*.
Two-Way Pager Address	The address for the two-way pager server.

In addition to defining the access points for the OracleAS Wireless site, you can configure the site's locale and time zone using the lists in the *Site Locale* section.

Figure 3–8 Configuring the Entry Points for Basic Site Configuration


ORACLE Enterprise Manager 10g
Application Server Control
Logs Topology Preferences Help
Farm > Application Server: as1012_041111_1600_mid.ias-pc2.us.oracle.com > Wireless >
Cancel Back Finish

Basic Site Configuration: Entry Points and Site Locale

Voice Entry Point

Voice Access Phone Number: 1-800-555-5000

Async Listener Access Points

Email Address: ask2@seagull.us.oracle.com

SMS Phone Number: 15555551234
(Example: 18001234567)

Instant Messaging Address: jabberfoo@jabber.org
Use the format of <network>|<User ID>. (Example: jabber|foo@jabber.org, yahoo|foo, msn|foo@msn.com, aim|foo, and icq|12345)

Two-Way Pager Address: 15555552345
(Example: 180012343567 or 180001234567@foo.com or 1800123.4567)

Site Locale

Locale: English

Time Zone: (UTC+00:00) Universal Time

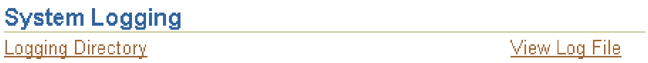
Copyright © 1996, 2004, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Application Server Control

After you complete this wizard, the configuration status in the *General* section displays as *Configured*.

3.4 Setting the Logging

From the *System Logging* section of the *Home* page (pictured in Figure 3–9), you can designate the location for the system logging and view the system log file.

Figure 3–9 The System Logging Section of the Home Page



3.4.1 Configuring the Logging Directory for the OracleAS Wireless Site

To configure the logging directory:

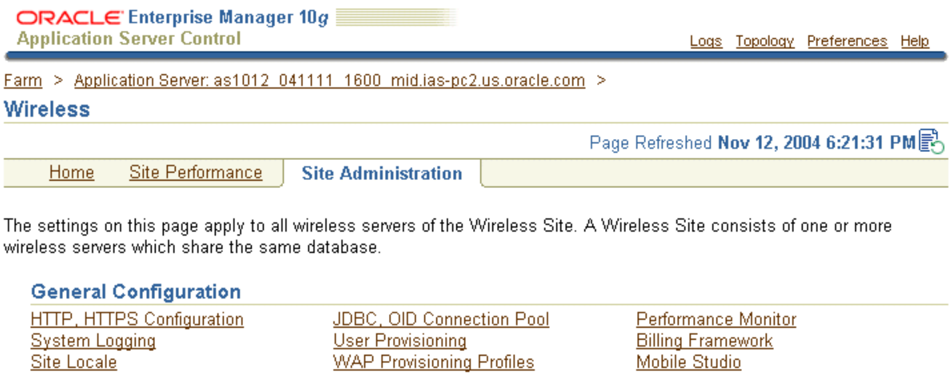
1. Click **Logging Directory**. The logging page appears.
2. Enter the name of the logging directory.
3. Click **OK**.

> **Note:** For changes to the logging directory to take effect, you must restart all of the OracleAS Wireless processes and the OC4J instances.

3.4.2 Configuring the System Logging for the OracleAS Wireless Site

From the *General Configuration* section (Figure 3–10) on the *Site Administration* page, you can change the log level for the whole site using the configuration page accessed by clicking **System Logging**.

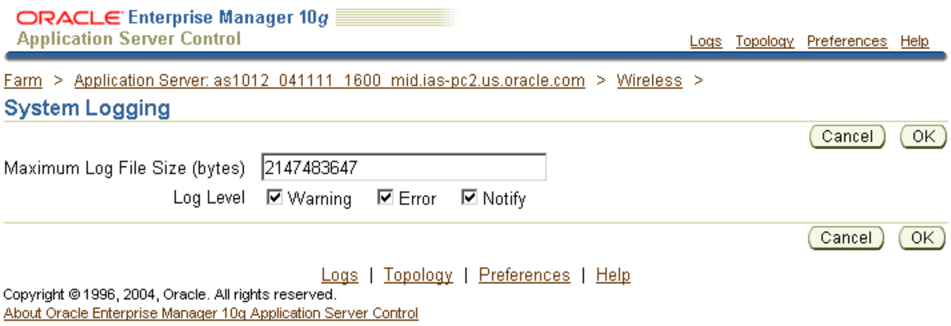
Figure 3–10 Accessing the System Logging from the Site Administration Page


ORACLE Enterprise Manager 10g
Application Server Control
Logs
Topology
Preferences
Help
Farm > Application Server: as1012_041111_1600_mid.ias-pc2.us.oracle.com >
Wireless
Page Refreshed Nov 12, 2004 6:21:31 PM
Home
Site Performance
Site Administration
General Configuration
HTTP, HTTPS Configuration
System Logging
Site Locale
JDBC, OID Connection Pool
User Provisioning
WAP Provisioning Profiles
Performance Monitor
Billing Framework
Mobile Studio

The settings on this page apply to all wireless servers of the Wireless Site. A Wireless Site consists of one or more wireless servers which share the same database.

From the *System Logging* page (Figure 3–11), you specify the log file size (in bytes), and select the logging level (*Error*, *Warning*, and *Notify*). By default, the *Error*- and *Warning*-level messages are written to the system log file.

Figure 3–11 The System Logging Page


ORACLE Enterprise Manager 10g
Application Server Control
Logs
Topology
Preferences
Help
Farm > Application Server: as1012_041111_1600_mid.ias-pc2.us.oracle.com > Wireless >
System Logging
Cancel
OK
Cancel
OK
Logs | Topology | Preferences | Help
About Oracle Enterprise Manager 10g Application Server Control

Maximum Log File Size (bytes) 2147483647

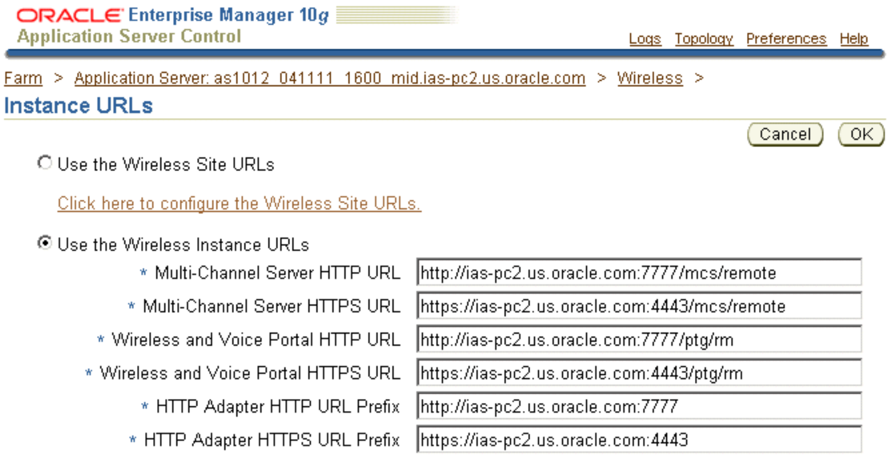
Log Level ☑ Warning ☑ Error ☑ Notify

Copyright © 1996, 2004, Oracle. All rights reserved.

3.5 Configuring the URLs of the Current OracleAS Wireless Instance

From the *Instance URLs* page (depicted in Figure 3–12), you can specify the URLs used by a OracleAS Wireless middle-tier server that are the entry points to the OracleAS Wireless services. This page enables you to define the instance URLs (that is, the local URLs) that act as the entry points for a middle-tier server, or direct a middle-tier server to use the URLs defined for the entire OracleAS Wireless site.

Figure 3–12 Configuring the Instance URLs for a OracleAS Wireless Server



Farm > Application Server: as1012_041111_1600_mid.ias-pc2.us.oracle.com > Wireless >

Instance URLs

○ Use the Wireless Site URLs

Click here to configure the Wireless Site URLs.

● Use the Wireless Instance URLs

* Multi-Channel Server HTTP URL	http://ias-pc2.us.oracle.com:7777/mcs/remote
* Multi-Channel Server HTTPS URL	https://ias-pc2.us.oracle.com:4443/mcs/remote
* Wireless and Voice Portal HTTP URL	http://ias-pc2.us.oracle.com:7777/ptg/rm
* Wireless and Voice Portal HTTPS URL	https://ias-pc2.us.oracle.com:4443/ptg/rm
* HTTP Adapter HTTP URL Prefix	http://ias-pc2.us.oracle.com:7777
* HTTP Adapter HTTPS URL Prefix	https://ias-pc2.us.oracle.com:4443

3.5.1 Defining the Instance URLs

Selecting *Use the Instance URLs* instructs the OracleAS Wireless server to use the URLs defined on this page, which are populated by the post-installer to enable OracleAS Wireless to work out of the box.

After completing the installations for each OracleAS Wireless server on the OracleAS Wireless site, you then configure the URLs for the OracleAS Wireless site as virtual URLs and then select the *Use the Wireless Site URLs* option for each of the OracleAS Wireless servers. When upgrading the OracleAS Wireless site, select this option for each server until all of the servers on the OracleAS Wireless site have been upgraded. See Section 3.9.1.1 for information on setting the URLs for the OracleAS Wireless site.

The instance URLs include those described in Table 3–3.

Table 3–3 The Instance URLs

Parameter	Value
Multi-Channel Server HTTP URL	The Multi-Channel Server URL in HTTP mode. This URL is used when the OracleAS Wireless server uses the Multi-Channel server entry point for URL re-writing. The default URL format is: *http://<server>:<http port>/mcs/remote*
Multi-Channel Server HTTPS URL	The Multi-Channel Server URL in HTTPS mode. The default URL format is: *https://<server>:<https port>/mcs/remote*
OracleAS Wireless and Voice Portal HTTP URL	The OracleAS Wireless and Voice Portal URL in HTTP mode. The default URL format is: *http://<server>:<http port>/ptg/rm*
OracleAS Wireless and Voice Portal HTTPS URL	The OracleAS Wireless and Voice Portal URL in HTTPs mode. The default URL format is: *https://<server>:<https port>/ptg/rm*

Table 3–3 (Cont.) The Instance URLs

Parameter	Value
HTTP Adapter HTTP URL Prefix	The URL prefix for the remote JSP page that is invoked by the HTTP Adapter in HTTP mode. Entering the URL prefix enables the OracleAS Wireless server to automatically attach this prefix to a JSP entered in the *Input Parameters* page of the Service Manager's *Master Application Creation Wizard*. When entering a JSP value in this wizard, you need only enter the JSP. For example, if you enter a remote JSP called *myApp.jsp*, into the wizard, the OracleAS Wireless server attaches the URL prefix, making this value into *http://remote_host:port/apps/myApp.jsp*. The default format is: *http://<server>:<http port>*
HTTP Adapter HTTPS URL Prefix	The URL prefix for the remote JSP page that is invoked by the HTTP Adapter in HTTPS mode. The default URL format is: *https://<server>:<https port>*
OracleAS Wireless Tools URL	The URL for the OracleAS Wireless Tools, which must be configured to enable the functioning of the utilities on the *Site Administration* page of the System Manager. The default URL is: *http://<server>:<port>/webtool*
OracleAS Wireless Customization Portal URL	The URL for the OracleAS Wireless Customization Portal. The default URL format is: *http://<server>:<port>/mobile*
J2ME Provisioning Server URL	A user's device is redirected to this URL when the user opts to download a J2ME application. The default URL format is: *http://<server>:<port>/provisioning/sun-ota*
J2ME Web Service Proxy Server URL	The URL to the proxy server that makes the Web services available to the J2ME applications built using the J2ME Web Services Client Library. The default URL format is: *http://<server>:<port>/mcs/wsproxy/proxy*
XMS Center Base URL	The URL to the MM1 entry point for the XMS Center. The default URL format is: *http://<server>:<port>/xms/mm1*
Audio Library URL Prefix	The HTTP root to the audio files for catspeech (concatenated speech). For example, if you set this to *http://localhost:7777/audio/catspeech*, then the catspeech server expects all audio files associated with its libraries to originate from that location. If this is set incorrectly, then no audio associated with catspeech plays; only TTS (text-to-speech) plays back. The default URL format is: *http://<server>:<port>/audio/catspeech*
Image Server HTTP URL	The URL to the Multimedia Adaptation service's image adaptation servlet (in HTTP mode). *http://<server>:<http port>/mcs/media/image*
Image Server HTTPS URL	The URL to the Multimedia Adaptation service's secure image adaptation servlet (in HTTPS mode). *http://<server>:<https port>/mcs/media/image*
Voice Grammar Server URL	The URL to the Multimedia Adaptation service's voice grammar adaptation servlet. The default URL format is: *http://<server>:<port>/mcs/media/vgrammar*

3.6 Managing the OracleAS Wireless Processes

From the *Home* page of the System Manager, you can manage the wireless processes on the local middle tier. There are two types of OracleAS Wireless processes:

- Web-based—The Web-based OracleAS Wireless application runs in a Wireless OC4J (OracleAS Containers for J2EE) instance. For more information, see Section 3.6.1.
- Standalone—The standalone Java processes can be started or stopped individually. For more information, see Section 3.6.2.

3.6.1 Managing and Configuring the Web-Based Applications

When you click the *Home* tab, the Web-based applications display the following types of OC4J applications running in Wireless OC4J instance, with name and status information:

- Multi-Channel Server
- Async Listener
- J2ME Web Service Proxy Server
- Multimedia Adaptation Server
- Provisioning Server
- Wireless Tools
- Customization Portal

If the application name appears as a link, then you can access a detail page that displays the application's performance information. You can start or stop all the Web-based applications by clicking the **Start OC4J Instance** or **Stop OC4J Instance** buttons.

3.6.2 Managing and Configuring Standalone Processes

The following OracleAS Wireless standalone processes display by name, running status and *Enabled* flag:

- Notification Engine
- Notification Event Collector
- Data Feeder
- Messaging Server
- Performance Monitor
- Location Event Server
- Oracle Sensor Edge Server

By selecting a process, you can start or stop it as well as enable or disable it.

> **Note:** You can only stop (and start) a process that has been enabled.

3.6.2.1 Creating a Standalone Process

Clicking *Add Process* invokes a two-step wizard that enables you to create a new process by first selecting the process type and entering the basic information about the process (such as the name) and then entering information specific to the process type. You can also select an existing process and delete it.

> **Note:** Names for the standalone processes can only contain single-byte, alphanumeric characters.

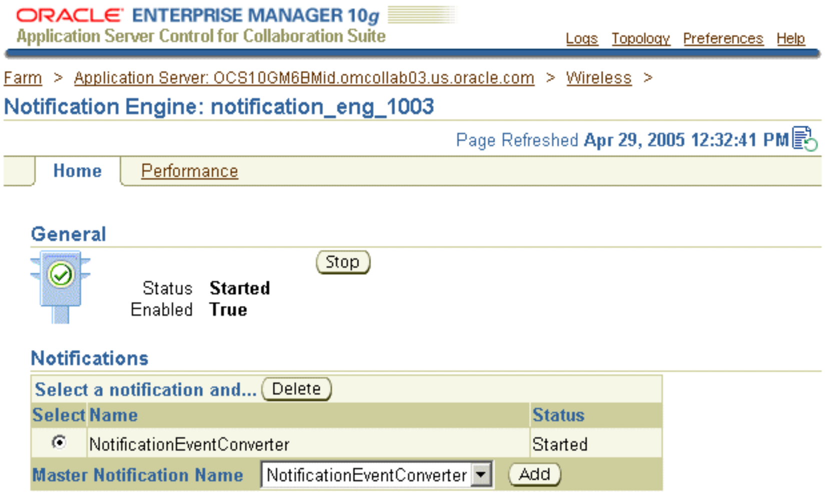
3.6.2.2 Configuring and Managing a Standalone Process

From the detail page, which you access by clicking the process name link, you can configure, or view the detail status and performance information of a standalone process. You can also start or stop the process from this page.

Notification Engine

You can configure the notification applications running in the process, and view their performance for notifications that are both processed and sent, subscribers to the notifications, and errors.

Figure 3–13 The Detail Page for a Notification Engine Process


ORACLE ENTERPRISE MANAGER 10g
Application Server Control for Collaboration Suite
Logs
Topology
Preferences
Help
Farm > Application Server: OCS10GM6BMid.omcollab03.us.oracle.com > Wireless >
Home
Performance
Stop
Delete
Add

Notification Engine: notification_eng_1003

Page Refreshed Apr 29, 2005 12:32:41 PM

General

Status **Started**
Enabled **True**

Notifications

Select a notification and...

Select	Name	Status
●	NotificationEventConverter	Started

Master Notification Name: NotificationEventConverter

Notification Event Collector

You can specify the components which process the notification events.

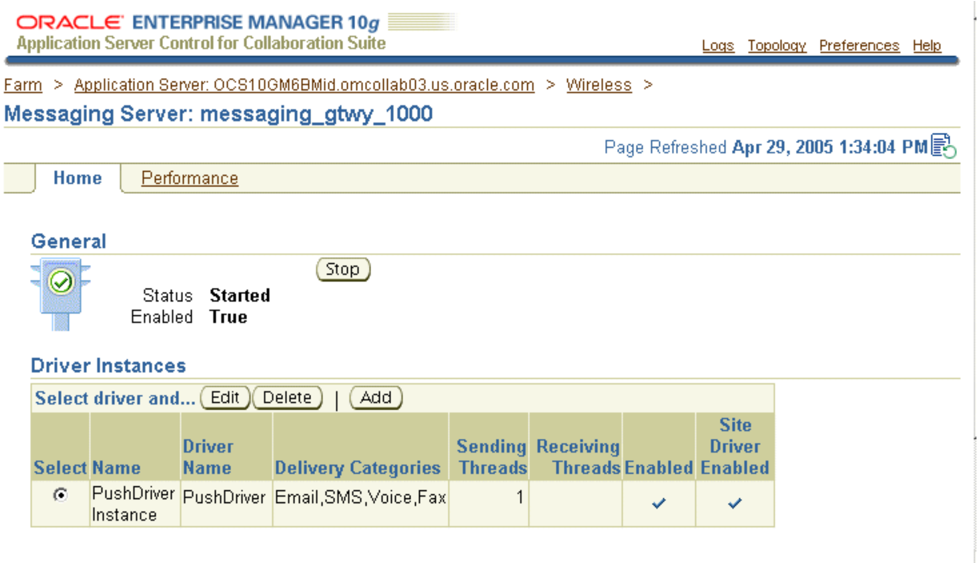
Data Feeder

You can configure the data feeders running in the process.

Messaging Server

You can configure the driver instances running in the process, which determine the type of messaging services by transport type. You can also view the OracleAS Wireless Server performance, such as the sending processing time, receiving response time and, number of messages sent for each delivery type. For more information on Messaging Server process performance, see Section 3.8.

Figure 3–14 The Detail Page for a Messaging Server Process



ORACLE ENTERPRISE MANAGER 10*g*
Application Server Control for Collaboration Suite

Logs Topology Preferences Help

Farm > Application Server: OCS10GM6BMid.omcollab03.us.oracle.com > Wireless >

Messaging Server: messaging_gtwy_1000

Page Refreshed **Apr 29, 2005 1:34:04 PM**

General

Status **Started**
Enabled **True**

Driver Instances

Select driver and... Edit Delete | Add

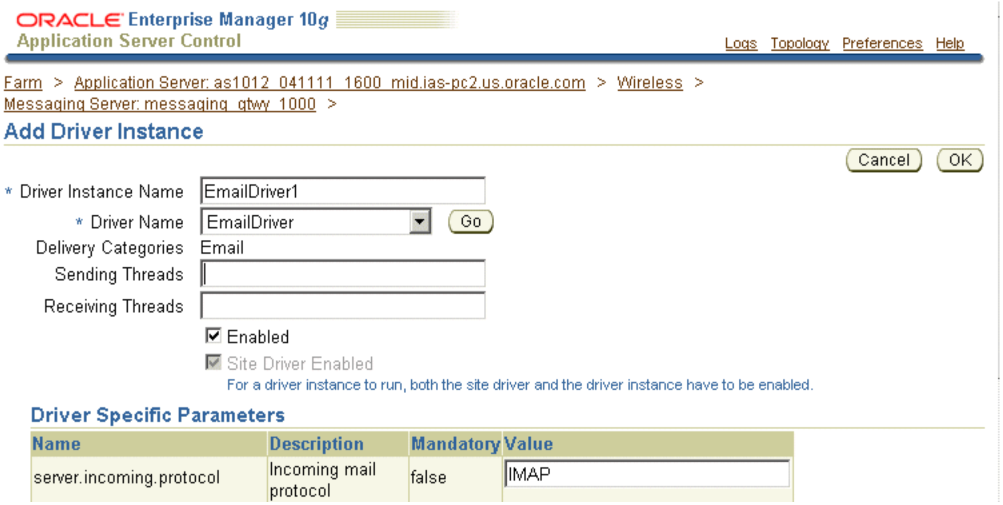
Select	Name	Driver Name	Delivery Categories	Sending Threads	Receiving Threads	Enabled	Site Driver Enabled
●	PushDriver Instance	PushDriver	Email,SMS,Voice,Fax	1		✓	✓

For the Messaging Server to function, you must configure the Messaging Server drivers at the site level and the Messaging driver instances at the Messaging Server process level.

3.6.2.2.1 Configuring the Messaging Drivers for the OracleAS Wireless Site Drivers are defined at site level under the *Messaging* category (located in the *Component Configuration* section). Each driver configuration includes category, capability (*Send*, *Receive* or *Both*), and driver class. For more information, see Section 3.10.3.1.

3.6.2.2.2 Configuring a Messaging Driver for a Messaging Server Process OracleAS Wireless enables you to specify the driver instances used by a Messaging Server process by using the *Add Driver Instance* page (Figure 3–15), which is invoked by clicking the **Add Driver Instance** button in the process detail page. This page enables you to create a new driver. To create a driver instance, enter a name for the instance, select the site-level driver on which the driver instance is based and then click **Go**.

Figure 3–15 The Add Driver Instance Page



ORACLE Enterprise Manager 10*g*
Application Server Control

Farm > Application Server: as1012_041111_1600_mid.ias-pc2.us.oracle.com > Wireless >
Messaging Server: messaging_gtwy_1000 >

Add Driver Instance

* Driver Instance Name	EmailDriver1
* Driver Name	EmailDriver ▾ Go
Delivery Categories	Email
Sending Threads	
Receiving Threads	
	☑ Enabled
	☑ Site Driver Enabled

For a driver instance to run, both the site driver and the driver instance have to be enabled.

Driver Specific Parameters

Name	Description	Mandatory	Value
server.incoming.protocol	Incoming mail protocol	false	IMAP

These driver instances, which you define at the process level, are based upon drivers defined for the entire OracleAS Wireless site. (That is, they are defined at the site-level.) You can create multiple driver instances based on the same site-level driver, each with different parameter values. As a result, different driver instances use the same driver class to send and receive messages, but the driver parameters have values specific to the driver instances themselves. For example, you can create two instances of EmailDriver that run simultaneously, but are connected to a different e-mail server by defining two different values for the EmailDriver parameters, `server.outgoing.host and server.incoming.host`. The attributes of a driver instance, which you define in the *Add Driver Instance* page, are as follows:

- **Driver Instance Name** - The driver instance name.
- **Driver Name** - The site level driver on which this driver is based. You can select from any of the drivers defined at the site level.
- **Number of Sending Threads** - The number of sending threads used by this driver. This field only displays for drivers with either the SEND or BOTH capability. If you do not enter a value, then the default value specified for the site-level driver is used instead. If you set the parameter value to 0, then the driver instance cannot messages. For more information, see *Drivers* in Section 3.10.3.1.
- **Number of Receiving Threads** - The number of receiving threads used by this driver. This field only displays for drivers with either the *RECEIVE* or *BOTH* capability. If you do not enter a value, then default value specified for the site-level is used instead. If you set the parameter value to 0, then the driver instance cannot receive messages. For more information, see *Drivers* in Section 3.10.3.1.
- **Enabled** - By selecting this flag, you enable the driver instance; otherwise, the instance is disabled if you do not set this flag. For a driver instance to run, both the site and process levels must be enabled. At the process level, OracleAS Wireless displays both site-level **Enable/Disable** flag and the process-level flag.
- **Site Enabled** - The value displayed (which is read-only from the driver instance page), states whether the site driver has been enabled for the site. You can edit this value by editing the site-level driver. For more information, see *Drivers* in Section 3.10.3.1.

Notes:

All of the pre-seeded site-level drivers are enabled by default. To improve performance, you can disable unneeded drivers.

For the Messaging Server function properly (that is, to send many messages), you must also configure such messaging driver instance class parameters as *username* and *password*.

- **Driver class parameters** - You define these parameters to specify the driver class parameter values. Each parameter has multiple attributes which are defined at the site level, such as parameter name, description, mandatory flag (displayed as *True* or *False*) and parameter value. Although the driver table in this page displays all of the driver's site-defined attributes, you can only specify the parameters values at the process level (their default values are set at the site level). For a mandatory parameter, you must provide a value to successfully create or update a driver instance. If you do not define a mandatory parameter, then OracleAS Wireless generates an error.

3.6.2.2.3 Updating a Messaging Driver Instance To update the driver instance, you select the driver instance from the Messaging Server process detail page and then click **Edit**.

Note: If you base a driver instance upon a driver whose parameters have changed (for example, from the addition of a new parameter with default value or the removal of an obsolete parameter), then OracleAS Wireless reflects these changes in the table listing the parameters in the editing page. If a new parameter has been added to the driver, then the table includes the new parameter with its default value. The table does not display parameters that have become obsolete and have been deleted.

Performance Monitor

You can configure the number of working threads for a Performance Monitor process.

Oracle Sensor Edge Server

To create an Oracle Sensor Edge Server process, enter a name for the process and then the name of the edge server group. The devices (and their filter instances) as well as the dispatchers, are assigned to an Oracle Sensor Edge Server process using the Sensor Services tool. For more information, see Chapter 4, "Managing Oracle Sensor Edge Services".

Note: If you modify the assignments of devices, device groups, or dispatchers of an Oracle Sensor Edge Server process, or modify the properties of the devices, device groups or dispatchers used by a process, then you must stop and restart the Oracle Sensor Edge Server process to which these components belong.

Location Event Server

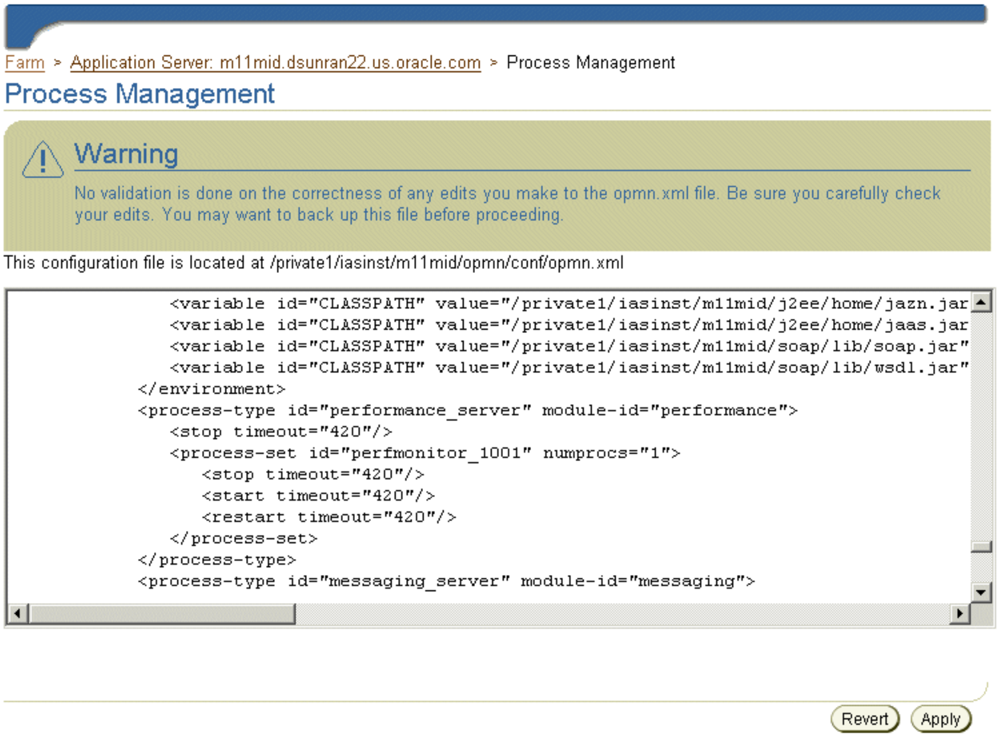
To configure a location event server process, you enter the number of positioning schedulers. Each location event server can have one or more positioning schedulers that process the location-based conditions. This setting specifies the number of positioning schedulers for each location event server. You base this setting on the system workload. If many location based-conditions are created and processed, then you should enter a number greater than 1 (such as 5 or 10).

However, if few location based-conditions are created and processed, one positioning scheduler will suffice. You can adjust this value according to the performance of the location event server.

3.6.2.3 Editing opmn.xml to Adjust the Timeout for Starting or Stopping a Standalone Process

By default, the timeout to start or stop a standalone process is 420 seconds. You can adjust this value by updating *opmn.xml (the main configuration file of OPMN, Oracle Process Management and Notification)* from the *Process Management* page. You can invoke this page (Figure 3–16) from the *Process Management* link of the application server page in the Enterprise Manager. All of the OracleAS Wireless standalone processes are listed under the OracleAS Wireless component in *opmn.xml*.

Figure 3–16 Accessing ompn.xml Through the Process Management Page



Farm > Application Server: m11mid.dsunran22.us.oracle.com > Process Management

Process Management

Warning

No validation is done on the correctness of any edits you make to the opmn.xml file. Be sure you carefully check your edits. You may want to back up this file before proceeding.

This configuration file is located at /private1/iasinst/m11mid/opmn/conf/opmn.xml

```
            <variable id="CLASSPATH" value="/private1/iasinst/m11mid/j2ee/home/jazn.jar
            <variable id="CLASSPATH" value="/private1/iasinst/m11mid/j2ee/home/jaas.jar
            <variable id="CLASSPATH" value="/private1/iasinst/m11mid/soap/lib/soap.jar"
            <variable id="CLASSPATH" value="/private1/iasinst/m11mid/soap/lib/wsdl.jar"
         </environment>
         <process-type id="performance_server" module-id="performance">
            <stop timeout="420"/>
            <process-set id="perfmonitor_1001" numprocs="1">
               <stop timeout="420"/>
               <start timeout="420"/>
               <restart timeout="420"/>
            </process-set>
         </process-type>
         <process-type id="messaging_server" module-id="messaging">
```

3.7 Monitoring the Performance of the OracleAS Wireless Server

The *Response and Load* section displays the following Wireless statistics, which are an overview of the process performance metrics based on the last 10 minutes for the local middle tier:

- Number of Active Sessions

  The number of sessions which invoked applications in the last 10 minutes.

- Average Response Time (second)

  The average response time for applications invoked in the last 10 minutes

- Average Session Duration (second)

  The average session duration for sessions invoked applications in the last 10 minutes

- Number of Applications Invoked

  The total number of applications invoked in the last 10 minutes

- Number of J2ME Applications Downloaded

  The number of J2ME applications downloaded in the last 10 minutes

- Number of Notifications Sent

  The number of notifications sent in the last 10 minutes

- Number of Messages Sent

  The number of messages sent in the last 10 minutes

- Number of Messages Received

  The number of messages received in the last 10 minutes

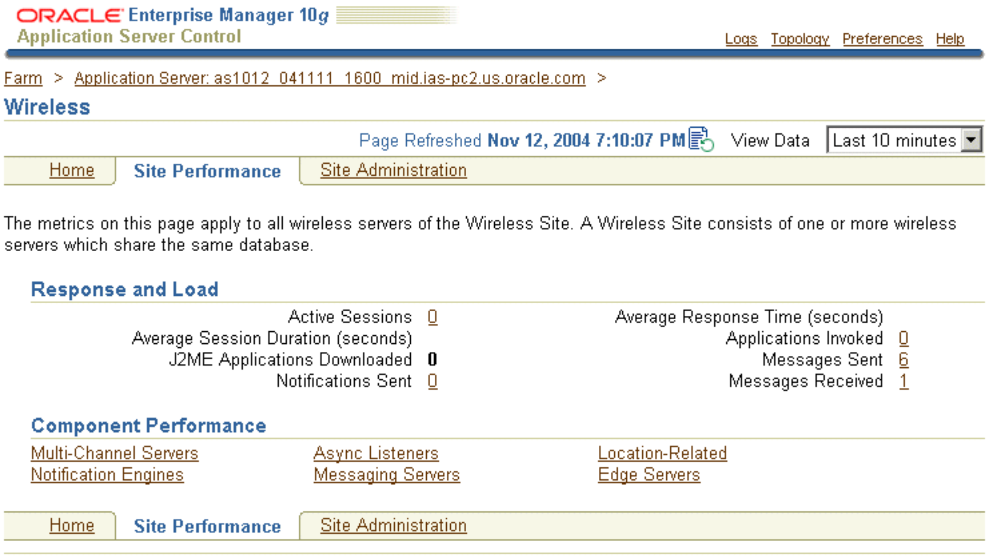
3.8 Monitoring the Performance of the OracleAS Wireless Site

On the *Site Performance* page (Figure 3–17), the *Response and Load* section displays the same type of performance data as the middle tier, but the data is for the entire Wireless Site. You can also select the *View Data* options for the time interval of the performance data. The choices are:

- Last 5 minutes
- Last 10 minutes (default selection)
- Last 30 minutes
- Last 60 minutes
- Last 1 day
- Last 7 days
- Last 31 days

> **Note:** You can select these time frame viewing options on any OracleAS Wireless performance page.

Figure 3–17 The Site Performance Screen (Partial View)



Page Refreshed Nov 12, 2004 7:10:07 PM View Data Last 10 minutes

The metrics on this page apply to all wireless servers of the Wireless Site. A Wireless Site consists of one or more wireless servers which share the same database.

Active Sessions	0	Average Response Time (seconds)	
Average Session Duration (seconds)		Applications Invoked	0
J2ME Applications Downloaded	0	Messages Sent	6
Notifications Sent	0	Messages Received	1

Clicking the links in the *Component Performance* section of the page enables you to view performance metrics within a selected time frame. The *Performance* page and the individual component performance pages each have a timestamp with a **Refresh** button, which enables you to reload the page to update the performance or status information.

Multi-Channel Server Performance

The performance data over the designated time period is displayed for each process of the OracleAS Wireless site. These performance categories include:

- Average Response Time (second)

  The average application response time for the specified period.

- Average Session Duration (second)

  The average duration for a session which invoked applications for the specified period.

- Number of Users

  The number of users who invoked applications for the specified period.

- Number of Applications Invoked

  The number of applications invoked for the specified period.

- Average Number of Application Invocations per Session

  The average number of application invocations for each session for the specified period.

- Average Number of Application Invocations per User

  The average number of application invocations for each user for the specified period.

- Number of Errors

  Total number of errors for the specified period.

Async Listener Performance

The performance data over the designated time period displays for each process of the OracleAS Wireless site. These performance categories include:

- Number of Messages Received

  The total number of messages received during the specified period.

- Average Message Response Time (second)

  The average processing time per message for the specified period.

- Average Message Queue Size

  The average message queue size during the specified period.

- Application Access Count

  The total number of applications accessed during the specified period.

- User Access Count

  The number of distinct users who accessed the site within the specified period.

- Number of Errors

  The total number of errors during the specified period.

Notification Engine Performance

The performance data over the designated time period will be displayed for each individual process of the OracleAS Wireless site:

- Number of Notifications Processed

  The total number of notifications processed for the specified period.

- Number of Notifications Sent

The total number of notifications sent for the specified time period.

- Number of Subscribers Notified

  The total number of users who received notifications during the specified time period. A subscriber is a user who accesses a notification (and sets trigger conditions for a notification).

- Number of Application Invocations

  The total number of application invocations over the specified time period. In this version of OracleAS Wireless, the notification message content is generated by invoking an application.

- Number of Errors

  The total number of errors occurred for the specified time period.

Messaging Server Performance

The performance data are separated by client-side performance and server-side performance. The client performance is based on the designated time period for each delivery type of each process of the OracleAS Wireless site:

- Average Sending Response Time (ms)

  The average time of a sending method. On the client side, a sending method is called to send a message. This time is the period from when the method is called to when the method returns. When the method returns, the message is saved in a database persistently, but is not delivered.

- Total number of Sending Requests

  The total number of times that the client process calls the sending method. The sending method can be called once to send a message to a set of destinations.

- Total Number of Sending Requests Sent

  The total number of successful calls, where a message is delivered to a proper gateway and its receipt is acknowledged. The client process can call the sending method many times to send many messages. Some of these requests fail, as in the case where a destination cannot be reached. Other requests could be pending.

- Total Number of Sending Requests Failed

  The total number of all calls that are known to have failed.

- Average Receiving Processing Time (ms)

  The average time taken by the messaging system to deliver a received message to the client.

The server performance is based on the designated time period for each delivery type of each process of the wireless site. These performance categories include:

- Average Sending Processing Time (ms)

  The average time taken by messaging system to send a message, starting from the sending method called by the client, to the driver delivered the message to the proper gateway.

- Average Receiving Response Time (ms)

  Once a transport driver receives a message, the message is passed to the transport system by an `onMessage` method. The response time is the time taken by the `onMessage` method. Once the `onMessage` returns, the received message is saved in a database for dispatching.

- Total Number of Received Messages

  The total number of times the transport drivers call the `onMessage` call-back method.

- Total Number of Received Messages Dispatched

  The total number of received messages which are dispatched to, and are accepted by, the listeners. Among received messages, some may be in processing. Others may not have been dispatched to the listeners, or the listeners may have failed to process the dispatched messages.

- Total Number of Received Messages Dispatched Failed

  The total number of received messages which failed to dispatch to a listener.

Location-Related Performance

The location related performance metrics are measured by location-based service provider and by location event server.

- Location-Based Service Provider

  These metrics display by *Provider Name* (the name of the application provider) and by *Provider Type* (the fully qualified class name associated with the provider) as follows:

  - Hits

    The number of times an attempt was made to use this provider. It includes both successful and unsuccessful attempts.

  - Average Success Rate

    The percentage of the times that a hit resulted in a connection to the provider and the return of structurally acceptable information.

  - Average Elapsed Time (ms)

    The average number of milliseconds that it took for a hit to have a successful or unsuccessful result.

- Location Event Server

  These metrics display by process name (the name of a location event server) process as follows:

  - Average Dequeue Time (seconds)

    The average number of seconds that elapsed between the time a request was ready in the queue and the time the dequeuing of the request was finished.

  - Average Evaluation Time (seconds)

    The average number of seconds that elapsed between the time the dequeuing of the request was finished and the time the result was generated. The result can be a determination of whether the condition is satisfied or not, or it can be an error.

Oracle Sensor Edge Server Processes Performance Data

For the Edge Server processes, the performance metrics display by process name as follows:

- Events Processed

  The total number of events processed for this process during the selected time period.

- Errors

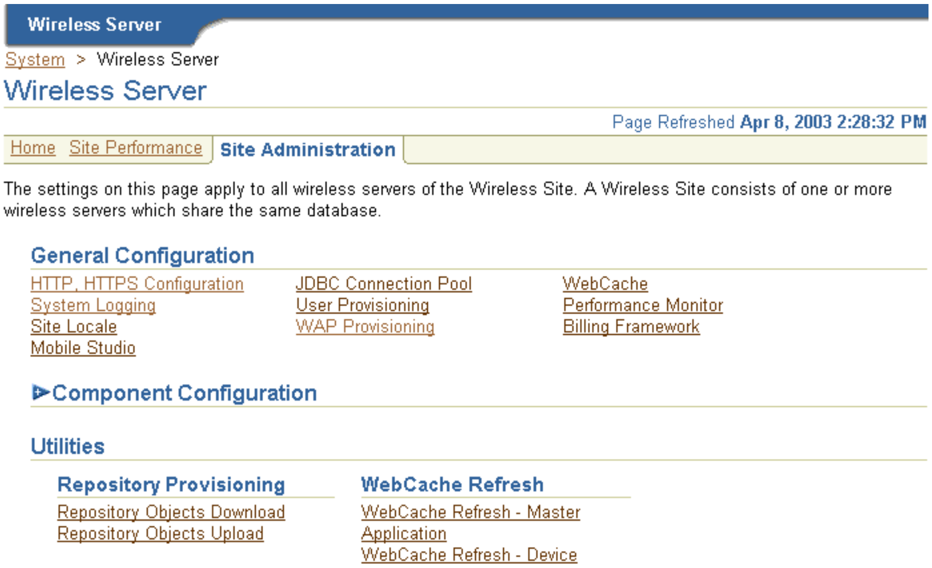
  The total number of errors encountered during the specified time frame.

3.9 Setting the Generic Configuration for the OracleAS Wireless Site

From the *Site Administration* page (Figure 3–18), you can configure the OracleAS Wireless system for the whole OracleAS Wireless site; all of the OracleAS Wireless servers use this common configuration. From this page, you can also access functions to download or upload repository objects.

The timestamp on the *Site Administration* page displays the last time that the configuration data was loaded from the database. To update the data on the page, click the **Refresh** icon or update some configuration data. Otherwise, the timestamp reflects the last time that the configuration data was loaded from the database.

Figure 3–18 The Site Administration Screen (Partial View)



Wireless Server

System > Wireless Server

Wireless Server

Page Refreshed **Apr 8, 2003 2:28:32 PM**

Home Site Performance **Site Administration**

The settings on this page apply to all wireless servers of the Wireless Site. A Wireless Site consists of one or more wireless servers which share the same database.

General Configuration

HTTP, HTTPS Configuration
System Logging
Site Locale
Mobile Studio

JDBC Connection Pool
User Provisioning
WAP Provisioning

WebCache
Performance Monitor
Billing Framework

▶Component Configuration

Utilities

Repository Provisioning

Repository Objects Download
Repository Objects Upload

WebCache Refresh

WebCache Refresh - Master
Application
WebCache Refresh - Device

3.9.1 General Configuration

The *General Configuration* section contains the generic configurations for the OracleAS Wireless system. These configurations include:

- Section 3.9.1.1, "HTTP, HTTPS Configuration"
- Section 3.9.1.2, "JDBC Connection Pool and OID Connection Pool"
- Section 3.9.1.3, "System Logging"
- Section 3.9.1.4, "Site Locale"
- Section 3.9.1.5, "User Provisioning"
- Section 3.9.1.6, "Virtual Users"
- Section 3.9.1.7, "WAP Provisioning"

- Section 3.9.1.8, "Performance Monitor"
- Section 3.9.1.9, "Billing Framework"
- Section 3.9.1.10, "Mobile Studio"

3.9.1.1 HTTP, HTTPS Configuration

The *HTTP, HTTPS Configuration* page enables you to configure the OracleAS Wireless site's proxy server settings, URLs, and the Secure Socket Layer (SSL) certificates.

3.9.1.1.1 Configuring the Proxy Server for HTTP The proxy server section enables you to configure the proxy properties used by Wireless for the HTTP protocol. If your network uses a proxy server, then you must set these properties to enable the proper functioning of such components as provisioning server, geocoding, and the XMS center.

> **Note:** If the OracleAS Wireless system does not use an HTTP proxy server, then you do not need to configure the proxy server properties.

To configure the proxy server:

- Specify the name proxy server host (such as *www-proxy.us.oracle.com*).
- Enter the proxy port number of the HTTP proxy server. The default port number is 80.
- Enter the exception addresses, which are addresses that do not require a proxy. The default setting is *localhost|127.01.0.1*. Separate the entries with a pipe bar (|).
- If the proxy server requires authentication, select **Proxy Server Requires Authentication**. If you select this option, then you must also provide a user name and password.

3.9.1.1.2 Configuring the URLs for the OracleAS Wireless Site This page enables you to also define the URLs for the site. These URLs can be used as the virtual URLs for OracleAS Wireless servers. To enable the URLs defined in this page (described in Table 3–4), select *Use the Site URLs* in the *Instance URLs* page. If you do not select this option, then the OracleAS Wireless servers use their local URLs instead. In addition to these URLs, you must define the following:

- The host name of the OracleAS Wireless server.
- The host port of the OracleAS Wireless server. The default is *7777*.
- The HTTP port of the OracleAS Wireless. The default is *4443*.
- The Audio Library URL prefix, such as *http://localhost:7777*.
- The live update URL prefix.
- The voice access number.

For more information on the *Instance URLs* page, see Section 3.5.1.

Table 3–4 The URLs for the Multi-Channel Server

Parameter	Value
Multi-Channel Server HTTP URL	The Multi-Channel Server URL in HTTP mode. This URL is used when the OracleAS Wireless server uses the Multi-Channel server entry point for URL re-writing. The default URL format is: *http://<server>:<http port>/mcs/remote*
Multi-Channel Server HTTPS URL	The Multi-Channel Server URL in HTTPS mode. The default URL format is: *https://<server>:<https port>/mcs/remote*
OracleAS Wireless and Voice Portal HTTP URL	The OracleAS Wireless and Voice Portal URL in HTTP mode. The default URL format is: *http://<server>:<http port>/ptg/rm*
OracleAS Wireless and Voice Portal HTTPS URL	The OracleAS Wireless and Voice Portal URL in HTTPs mode. The default URL format is: *https://<server>:<https port>/ptg/rm*
HTTP Adapter HTTP URL Prefix	The URL prefix for the remote JSP page that is invoked by the HTTP Adapter in HTTP mode. Entering the URL prefix enables the OracleAS Wireless server to automatically attach this prefix to a JSP entered in the Input Parameters page of the Service Manager's Master Application Creation Wizard. When entering a JSP value in this wizard, you need only enter the JSP. For example, if you enter a remote JSP called *myApp.jsp* into the wizard, the OracleAS Wireless server attaches the URL prefix, making this value into *http://remote_host:port/apps/myApp.jsp*. The default format is: *http://<server>:<http port>*
HTTP Adapter HTTPS URL Prefix	The URL prefix for the remote JSP page that is invoked by the HTTP Adapter in HTTPS mode. The default URL format is: *https://<server>:<https port>*
OracleAS Wireless Tools URL	The URL for the OracleAS Wireless Tools, which must be configured to enable the functioning of the utilities on the Site Administration page of the System Manager. The default URL is: *http://<server>:<port>/webtool*
OracleAS Wireless Customization Portal URL	The URL for the OracleAS Wireless Customization Portal. The default URL format is: *http://<server>:<port>/mobile*
J2ME Provisioning Server URL	A user's device is redirected to this URL when the user opts to download a J2ME application. The default URL format is: *http://<server>:<port>/provisioning/sun-ota*
J2ME Web Service Proxy Server URL	The URL to the proxy server that makes the Web services available to the J2ME applications built using the J2ME Web Services Client Library. The default URL format is: *http://<server>:<port>/mcs/wsproxy/proxy*
XMS Center Base URL	The URL to the MM1 entry point for the XMS Center. The default URL format is: *http://<server>:<port>/xms/mm1*

Table 3–4 (Cont.) The URLs for the Multi-Channel Server

Parameter	Value
Audio Library URL Prefix	The HTTP root to the audio files for catspeech (concatenated speech). For example, if you set this to *http://localhost:7777/audio/catspeech*, then the catspeech server expects all audio files associated with its libraries to originate from that location. If this is set incorrectly, then no audio associated with catspeech plays; only TTS (text-to-speech) plays back. The default URL format is: *http://<server>:<port>/audio/catspeech*
Image Server HTTP URL	The URL to the Multimedia Adaptation service's image adaptation servlet (in HTTP mode). *http://<server>:<http port>/mcs/media/image*
Image Server HTTPS URL	The URL to the Multimedia Adaptation service's secure image adaptation servlet (in HTTPS mode). *http://<server>:<https port>/mcs/media/image*
Voice Grammar Server URL	The URL to the Multimedia Adaptation service's voice grammar adaptation servlet. The default URL format is: *http://<server>:<port>/mcs/media/vgrammar*

3.9.1.1.3 Configuring SSL Certificates The OracleAS Wireless Server uses certificates to positively identify certification authorities and publishers. Before you can use HTTPS to connect to a content provider, you must download the content provider's SSL root certificates (see Section 16.5.21 for more information). After you download the certificate, copy it to the OracleAS middle tier.

The SSL section of this page enables you to configure the Wireless Server to use security certificates. OracleAS Wireless supports security certificates as either Base64 or PKCS#7-formatted certificate files to enable use of the HTTPS protocol. A Base64 certificate file is a text file, with the certificate information bounded at the beginning by '--BEGIN CERTIFICATE--' and at the end by '--END CERTIFICATE--'. A PKCS#7-formatted file is in binary code.

Note: If there is more than one middle tier, then you must copy the file to all middle tiers. The file must be in the same location on all of the middle tiers.

Once you have downloaded the certificate file to the middle tier, you must configure the OracleAS Wireless Server to use that certificate. If there is more than one middle tier, the you need only configure the first middle tier; all of the other middle tiers will use the same settings (that is why the certificate file must be in the same location on all middle tiers).

To configure the OracleAS Wireless to use a certificate that has been copied to the middle tier, enter the absolute path to the certificate file. You can set more than one certificate by clicking **Add Another Row**. You can also delete or update the certificate's file name. See also Section 10.4.2 and Section 16.5.22. For more information on SSL, refer to the Oracle Application Server Administrator's Guide and the Oracle Application Server Single Sign-On Administrator's Guide.

Note: You must configure the Secure Sockets Layer to use HTTPS in the HttpAdapter.

3.9.1.2 JDBC Connection Pool and OID Connection Pool

Pooling for JDBC connections improves resource utilization and reduces the connection establishment overhead when you access the database. The JDBC Connection Pool page, invoked by selecting the **JDBC, OID Connection Pool** link from the *Site Administration* page, enables you to configure the JBDC and OID (Oracle Internet Directory) connections for the site, including:

- Minimum number of connections (the default is 4).
- Maximum number of connections (the default is 10).
- Incremental allocation of new connections to the connection pool (the default is 1).
- Minimum number of connections to the OID connection pool.
- Maximum number of connections to the OID connection pool.

3.9.1.3 System Logging

For information, refer to Section 3.4.

3.9.1.4 Site Locale

The *Site Locale* page, invoked by selecting the Site Locale link from the *Site Administration* page, enables you to configure the locale and time zone for the site.

You can specify the default site locale and time zone. The default site locale can be selected from the list of all the supported locales of OracleAS Wireless. OracleAS Wireless ships with 29 supported locales which enable the translation of end-user messages into 29 languages. You can add a new locale or delete a locale using this page. For more information, see Section 15.2.4.

3.9.1.5 User Provisioning

The *User Provisioning* page enables you to set the properties used by the Provisioning adapter.

Table 3–5 describes the properties for normal user provisioning.

Table 3–5 User Provisioning Properties

Property	Description
Parent folder	The folder for the user's home folder. A new subfolder is created for every new user. The default is */Users Home*.
Default groups	The default group to which the user belongs. The default is *Users*. (You can select or clear the group selection using Control + click).
Disclose User Location	Selecting this option enables the users' location to be disclosed to a third-party application.
Disclose User Identity	Selecting this option enables the users' identities to be disclosed to a third party application.

3.9.1.6 Virtual Users

A virtual user is a user who accesses a OracleAS Wireless site, but does not register. When such a user accesses a OracleAS Wireless site, OracleAS Wireless detects the user and creates a virtual user account for that user.

Table 3–6 describes the properties for the virtual user provisioning.

Table 3–6 Virtual User Properties

Property	Description
Parent folder	The parent folder for the virtual user's home folder. A new subfolder is created for every new user. The default is */Users Home*.
Default groups	The default groups to which the user belongs. The default is *Users*. (You can select or clear the group selection using Control + click).
Enable Virtual User	Selecting this option enables a virtual user to create an account.

3.9.1.7 WAP Provisioning

You can create, edit, and delete WAP profiles using the *WAP Profile* page, which you access by selecting the **WAP Provisioning** link. This page displays a list of current WAP profiles. You can also add a WAP profile by defining the following parameters.

> **Note:** The parameters differ depending on the bearer technology that you select.

Table 3–7 describes the WAP provisioning profile parameters.

Table 3–7 WAP Provisioning Profiles

Parameter	Value
WAP Profile Name	The name of the WAP profile. You can name the profile for the WAP provider.
WAP Bearers	A list of the transport technologies.
GSM/CSD	Circuit-Switched Data (CSD) over a GSM (Global System for Mobile communication) network. This is the basic transfer protocol in GSM phones.
GSM/SMS	Short-Messaging Service over a GSM (Global System for Mobile communication) network. Select this store-and-forward technology to enable alphanumeric messaging between mobile phones and such other platforms as e-mail or voice mail.
GSM/USSD	Unstructured Supplementary Service Data (USSD) over a GSM (Global System for Mobile communication) network. USSD is both session- and transaction-oriented.
GPRS	General Packet Radio Service (GPRS). Select this bearer technology to use WAP on a per-transaction basis. GPRS enables services to be always on; a GPRS customer does not have to invoke a service to receive content.
WAP Gateway Proxy	The address of the WAP proxy server. For GPRS and GSM/CSD, it is an IP address. For GSM/SMS, this is service number or phone number. For GSM/USSD, this is either an IP address or an MSISDN number. This is a required field.
Port	The port number. The default port numbers are: ▪ 9200 (connection-less) ▪ 9201 (connection-oriented) ▪ 9202 (secure and connection-less) ▪ 9203 (secure and connection-oriented)

Table 3–7 (Cont.) WAP Provisioning Profiles

Parameter	Value
Secure WAP Session	Selecting this option enables WTLS (Wireless Transport Layer Security).
Phone Model	The brand and model of the wireless phone.
Access	The access point. For GPRS, this is the access point of the GPRS operator such as www.companyname.com. For GSM/CSD, the access point is a telephone number.
Home Page	The home page of the ISP provider accessed by the WAP user.
Call Type	A list of the call types (analog or ISDN) used for the connection.
Access	The access telephone number.
Call Speed	The call speed of the connection.
Authentication Type	Select one of the following protocols used for user authentication: ■ PAP (Password Authentication Protocol) ■ CHAP (Challenge Handshake Authentication Protocol).
ISP Name	The name of the Internet service provider (ISP).
ISP Login Name	The user name.
ISP Login Password	The user's password.
SMSC Address	The number of the SMSC (Short Message Service Center).
USSD Parameters	
Proxy Type	The phone number or IP address of the WAP provider.
USSD Service Code	The USSD code (for example, *555*), that precedes the destination number.

3.9.1.8 Performance Monitor

The Performance Monitor page enables you to configure the OracleAS Wireless performance monitor, including the parameters described in Table 3–8.

Table 3–8 Parameters of the Performance Monitor Page

Parameter	Description
Enable Performance Logging	Selecting this check box enables performance logging to the database.
Delimiter for logged name/value pair	The delimiter for the logged name/value pairs. The default delimiter is #%=%#. This is a required parameter.
Delimiter for logger records	The delimiter for the logged records. The default is *~#$*. This is a required parameter.
Wakeup Frequency (minute)	The number of minutes after which the logger thread wakes up to check for any new files in the process directory. The default is one minute. This is a required parameter.
Close Frequency (second)	The number of seconds to close a file. The default is *300*.
Batch Size for Performance logging	The batch size for the performance logging. The default is *15*. This is a required parameter.

3.9.1.9 Billing Framework

The *Billing Framework* page enables you to configure the OracleAS Wireless Billing Integration Framework, which provides an extensible and flexible framework to model billable services, capture billable action, and integrate with any external billing engine.

To enable the billing of all services, select *Enable Billing*. Billing is disabled by default.

To complete the process to enable billing, provide the implementation of two interfaces, the `BillingDataCollector` interface and the `BillingDriver` interface, and then configure them as the implementation classes.

Note: The out-of-the-box implementation of the `BillingDataCollector` interface is pre-seeded in the configuration as `oracle.wireless.billing.BillingDataCollectorImpl.`

The *Billing Collector Class,* fetches all of the component-specific billing attributes and then plugs them into the service detail record (SDR), which encapsulates the billable action. The *Billing Collector Class* considers the following components: Runtime, Notification Server, Provisioning Server, and Messaging Server.

In addition, you define the *Billing Provider Driver*, the driver implementation provided at the customer end which communicates with the external billing system. To enter this value, you enter the full class with the package name, such as `oracle.wireless.billing.SampleBillingDriver`.

You can select, delete, or add the driver class initialization `(init)` parameters. If this billing driver implementation class expects initialization properties, then you add them as name-value pairs.

For more information about billing framework, refer to the *Oracle Application Server Wireless Developer's Guide*.

3.9.1.10 Mobile Studio

The Mobile Studio page enables you to configure Mobile Studio by defining the parameters described in Table 3–9.

Note: You must restart the OracleAS Wireless server for the Mobile Studio configuration settings to take effect.

For more information on Mobile Studio, see Section 7.2 in Chapter 7, "Administering Mobile Studio".

Table 3–9 Parameters of the Mobile Studio Screen

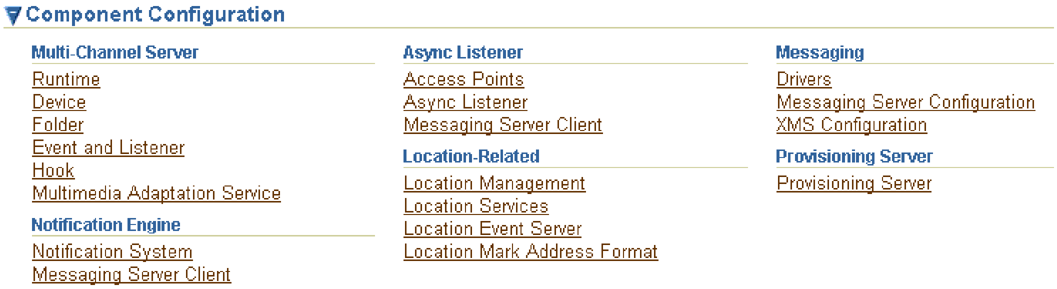
Parameter	Value
URL of Deploy Server	The URL of the OracleAS Wireless production instance. Applications created by developers in the Mobile Studio (referred to as the development instance) are deployed to this URL. For example, enter *http://myserver.mycompany.com:myport/studio*. If you do not enter the URL in this field, then deployment is disabled.
Default Site	The name of the branding (that is, the look and feel) which is used as the default. This is pre-seeded with the value *Default*. Application providers can brand the Mobile Studio (by customizing its appearance and content) and integrate it with an existing Web site. You can substitute another branding for this default by entering the name of the other branding in this field. For more information on branding, refer to the *Oracle Application Server Wireless Developer's Guide*.
J2ME Web Services Supported?	By selecting this option, Mobile Studio's interface displays an additional tab that includes functions that enable developers to register Web Services which can be accessed from J2ME MIDlets. By default, this option is not selected (the flag is set to *false*).

3.10 Configuring the OracleAS Wireless Components

The *Component Configuration* section (Figure 3–19) contains the configurations specific to different OracleAS Wireless components. To access the configuration pages for these components, you expand the *Component Configuration* section by clicking the plus (+) sign. The following sections describe the tasks enabled through the *Component Configuration* pages:

- Section 3.10.1, "Configuring the Multi-Channel Server"
- Section 3.10.2, "Configuring the Async Listener"
- Section 3.10.3, "Configuring Messaging"
- Section 3.10.4, "Configuring the Notification Engine"
- Section 3.10.5, "Configuring the Location-Related Components"
- Section 3.10.6, "Configuring the Provisioning Server"

Figure 3–19 The Component Configuration Section of the Administration Screen



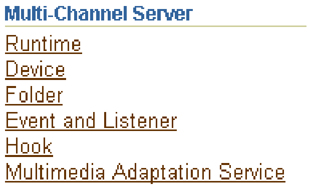
3.10.1 Configuring the Multi-Channel Server

The *Multi-Channel Server* component (Figure 3–20) section of the *Site Administration* page enables you to do the tasks described in Table 3–10.

Table 3–10 Multi-Channel Server Components and Related Tasks

Component Name	Task
Runtime	Configuring the Runtime
Device	Configuring Device ID Information
Folder	Configuring the Folder and Application Sorting Order
Event and Listener	Configuring the Request, Session, and Response Events
Hook	Changing the Hook Implementation Class
Multi-Media Adaptation Service	Overriding the Default Adaptation Services

Figure 3–20 The Multi-Channel Server Component



Multi-Channel Server

Runtime

Device

Folder

Event and Listener

Hook

Multimedia Adaptation Service

3.10.1.1 Configuring the Runtime

The *Runtime* page contains the configuration for runtime attributes, such as runtime session, and the object cache synchronization. Table 3–11 describes the runtime parameters.

Table 3–11 The Runtime Parameters

Parameter	Description
Runtime Session Life Time (seconds)	The life span of a session. The default is 600.
Runtime Session Check Interval (seconds)	The time required for the session monitor to check an open session. The default is 60.
Cache Object Life Time (seconds)	The life span of a persistent object. After this time, OracleAS Wireless reconstructs the object. The default is 600.
Cache Object Check Interval (seconds)	The time required for the cache monitor to check the cache. If the time is set to *-1*, Wireless does not invoke the cache monitor and the cache is not cleared. The default is 60 seconds.
Maximum execution time per Request (seconds)	OracleAS Wireless interrupts the threads for requests that take longer than this allotted time and returns an error. The default is 120 seconds.
Persistent Session Life Time (days)	The life span of a persistent session. Runtime session states include the state of user authentication, credentials, cookies, URL caches, the **short name**s for the asynchronous applications, and the module call-back stacks. Setting the Runtime Session Persistency flag makes these session states persistent.The lifetime of persistent sessions can be several orders of magnitude longer than the session expiration time. The default lifetime for a persistent session is two days.
Enable Runtime Session Persistency	Setting this flag enables a persistent session. The default is false.

For more information on the runtime, see *Oracle Application Server Wireless Developer's Guide*

Defining the parameters in the *Object Cache Synchronization* section of the page enables you to configure the thread pool, which handles the cache synchronization for messages. To configure the object cache synchronization, define the following parameters:

- Minimum number of threads in the thread pool.
- Maximum number of threads in the thread pool.
- Timeout, in minutes, for the threads in the thread pool.

3.10.1.2 Configuring Device ID Information

The *Device Configuration* page enables you to add, edit, or delete HTTP header names that contain information for the device ID. You can also configure the Multi-Channel Server setup menu with the following attributes:

- Enable Login
- Enable Logout
- Enable User Info
- Enable Service Customization
- Enable Global Preset
- Enable User Profile
- Enable Self-Registration
- Enable Home
- Enable Help (You must enter the URL of the help files if you select *Enable Help*.)

3.10.1.3 Configuring the Folder and Application Sorting Order

Using the *Folder* page, you can configure the folder sorting order and display by performing the following:

1. Selecting the sorting order for applications and folders on the output devices by using the arrows to select (> or >>) or remove (< or <<). The selection choices are ascending order or descending order (based on name), sequence number, or date:
   - ORDER_NAME_ASC
   - ORDER_NAME_DESC
   - ORDER_SEQNO_ASC
   - ORDER_SEQNO_DESC
   - ORDER_DATE_ASC
   - ORDER_DATE_DESC

   > **Note:** The ascending (ASC) or descending (DESC) sorting orders cannot be selected for the same property. For example, you cannot select both ORDER_NAME_ASC and ORDER_NAME_DESC.

2. Selecting the display application size under a folder, which is the number of applications to display in one folder.
3. Selecting from the following options for the user's home folder sorting policy:

- USE_ORDER_SERVICES (default value)
- USER_SERVICES_FIRST
- GROUP_SERVICES_FIRST
- Selecting the folder icon and audio settings

4. Configuring the URI for the icons, images and audio for folder, including *Generic Title Icon*, *Home Icon*, *Help Icon*, *Login Icon*,*Top Bar Image,* and *Help Audio*.

3.10.1.4 Configuring the Request, Session, and Response Events

The *Event and Listener* page displays the event options and available listeners. Using this page, you can enable or disable event generation by selecting from among the event options and listeners. You also use the page to add, update or remove a listener for the request events, session events, or response events.

The *Event and Listener* page includes the following configuration options for events (described in Table 3–12), which you can enable by selecting the appropriate check boxes. If you do not select an option, then the option is disabled (which is the default setting).

Table 3–12 The Request, Session, and Response Event Options

Option	Definition
Enable 'before request' Event	Declares a request event to be "just received".
Enable 'after request' Event	Declares a request event as "request object has been released".
Enable 'transform begin' Event	Declares an request event to be "before the transformation".
Enable 'request begin' Event	Declares a request event to"begin being processed".
Enable 'service begin' Event	Declares a request event to be "before the adapter is invoked".
Enable 'transform end' Event	Declares a request event to be "transformation complete".
Enable 'request end' Event	Declares a request event to be "request has been completely processed".
Enable 'service end' Event	Declares a request event to be "adapter execution complete".
Enable 'request error' Event	Declares a request event to be "error occurs during request processing."
Enable 'before session' Event	Declares a session event to be "before session starts".
Enable 'session authentication' Event	Declares a session event to be "session has been authenticated".
Enable "session begin" Event	Declares a session event to be "session has been validated".
Enable 'session end' Event	Declares a session event to be "session has expired (implicitly and explicitly)".
Enable 'after session' Event	Declares a session event to be "session object has been released".
Enable 'response error' Event	Declares a response event to be "error in response" object.

See the *Oracle Application Server Wireless Developer's Guide* for more information on event listeners.

3.10.1.5 Changing the Hook Implementation Class

Using the *Hooks* page, you change the hook implementation class for a selected hook (described in Table 3–13), which provides an extension mechanism for the Multi-Channel Server. You must implement the Java `Hook` interface specific to the type of intended extension or plug-in to OracleAS Wireless.

Table 3–13 Hooks for the Multi-Channel Server

Hook	Description
wireless.http.locator.signon.pages.hook.class	The hook to generate the sign-on page on the device. The default is `oracle.mwa.core.omap.panama.MWASignOnPage`.
wireless.http.locator.caller.location.hook.class	Declares the hook for which acquires the user's current location. The default is `oracle.panama.rt.common.LocAcq`.
wireless.http.locator.service.visibility.hook.class	Declares the hook for checking the show or hide status when Wireless starts. The default is `oracle.panama.rt.common.ServiceVisiblity`.
wireless.http.locator.listener.registration.hook.class	Declares the hook for the event registration listener. The default is `oracle.panama.rt.common.ListenerRegistration`.
wireless.http.home.folder.sorter.hook.class	Declares the hook for sorting a user home folder contact. The default is `oracle.panama.rt.common.HomeFolderSorter`.
wireless.http.locator.mobile.id.hook.class	Declares a hook to acquire a mobile ID. The default is `oracle.panama.rt.common.MobileIdHookImpl` .
wireless.http.locator.pre.processor.hook.class	Declares a hook to be invoked before device transformation.
wireless.http.locator.authorization.hook.class	Declares the hook for user service authorization. The default is `oracle.panama.rt.common.Authorizer`.
wireless.http.locator.post.processor.hook.class	Declares a hook to be invoked after device transformation.
wireless.http.locator.device.identifcation.hook.class	Declares the hook for identifying a device. The default is `oracle.panama.rt.hook.DeviceModels`.
wireless.http.locator.location.service.visibility.hook.class	Declares the hook to show or hide the contents of a folder based on its current location. The default is `oracle.panama.rt.hook.Folder.RendererPolicy` .
wireless.http.locator.folder.render.hook.class	Hook for a folder renderer. The default value is `oracle.panama.rt.common.FolderRenderer`.
wireless.http.locator.session.id.hook.class	Declares a hook for generating the session ID. The default is `oracle.panama.rt.common.SessionIDGenerator`.

Table 3–13 (Cont.) Hooks for the Multi-Channel Server

Hook	Description
wireless.http.locator.authentication.hook.class	Declares the hook for user authentication. The default is `oracle.mwa.core.omap.panama.OMAPAuthentication`.
wireless.http.locator.useragent.class	Default implementation of the device recognition class. The default is `oracle.panama.core.xform.UserAgentImpl`.
wireless.http.locator.normalizeaddress.hook.class	Stores the address field of `DeviceAddress` in normalized form, which is used to look up objects and to send the address by the transport. For example, the normalized form of an e-mail delivery type can be lower-case letters, making the normalized form of Scott.Tiger@Oracle.com into scott.tiger@oracle.com. The normalized form of the SMS delivery type could be all non-numeric characters. For example, the normalized form for (650) 555-5000 is 6505555000. If some carriers have a space between the area code and the rest of the number, then the normalized address logic converts the phone number to 650 555 5000.

3.10.1.6 Overriding the Default Adaptation Services

Multimedia adaptation services provide device-specific adaptation of images, ring tones, voice grammar, as well as audio and video streams. OracleAS Wireless provides the default implementation for these services. To use different implementations, change the corresponding provider class name on the *Multimedia Adaptation Service* configuration page.

> **Note:** When changing the class name, be sure that the class is in the OracleAS Wireless classpath.

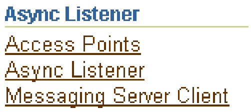
3.10.2 Configuring the Async Listener

For the Async Listener (Figure 3–21), you configure the Async-related components described in Table 3–14.

Table 3–14 Async LIstener Components and Related Tasks

Component Name	Task
Access Points	Managing Access Points
Async Listener	Configuring the Async Listener
Messaging Server Client	Configuring the Async Listener as a Client of the Messaging Server

Figure 3–21 The Async Listener Component



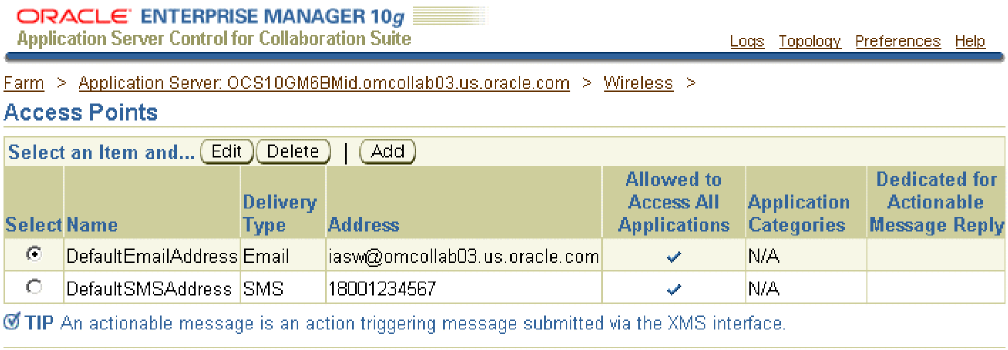
3.10.2.1 Managing Access Points

An access point is the address monitored by the Async Listener. Using the *Access Points* page, which lists the access points by the categories described in Table 3–15, you can add, delete, or edit a selected access point.

Table 3–15 Access Point Attributes

Attribute	Description
Name	The unique name for this access point.
Delivery Type	The delivery type for this access point address. Options include: *Mail*, *SMS*, *IM* or *Two-Way Pager*.
Address	The address for this access point. For SMS, it is a phone number, such as *18001234567*. For IM, it has the format of *<network>* | *<User ID>*, such as *jabber* | *foo@jabber.org*, *yahoo* | *foo*, *msn* | *foo@msn.com*, aim | foo, and *icq* | *12345*. OracleAS Wireless supports the Yahoo, MSN, AOL, ICQ, and Jabber networks. For two-way pagers, use the format *180012343567* or *180001234567@foo.com* or *1800123.4567*.
Allowed to Access All Applications	Select this option to determine if this is a site access point, or an application category access point. If you do not select this option, then you can associate one or more application categories with an access point used to support **Premium SMS**. If you select this option and create a site access point, then OracleAS Wireless prompts you to confirm the removal of all of the application categories (if any) associated with this access point. See also Section 3.10.2.2 for more information on site access points.
Dedicated for Actionable Message Reply	Selecting this option creates an address that is dedicated for **actionable message** reply. Once it is set, all of the actionable push messages have the *From* address set to the access point. The instructions for replying to an actionable message omit the **short name**. To answer these messages, users need only to reply with a transaction ID and the application parameters.
Application Categories	The categories associated with an application category access point. The field is read-only, and it only appears when you edit an access point. This field is populated with values only if you did not select the *Allowed to Access All Applications* option.

Figure 3–22 The Access Point Browsing Screen



ORACLE ENTERPRISE MANAGER 10g
Application Server Control for Collaboration Suite
Logs
Topology
Preferences
Help

Farm > Application Server: OCS10GM6BMid.omcollab03.us.oracle.com > Wireless >

Access Points

Select an Item and...
Edit
Delete
Add

Select	Name	Delivery Type	Address	Allowed to Access All Applications	Application Categories	Dedicated for Actionable Message Reply
●	DefaultEmailAddress	Email	iasw@omcollab03.us.oracle.com	✓	N/A	
○	DefaultSMSAddress	SMS	18001234567	✓	N/A	

TIP An actionable message is an action triggering message submitted via the XMS interface.

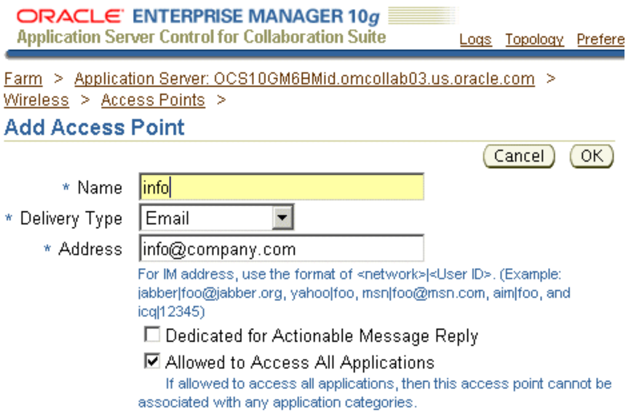
3.10.2.2 Creating Access Points

Clicking the **Add** button invokes the *Add Access Point* page (Figure 3–23). The access points that you create using this page depend on the parameters defined for the e-mail, IM, Two-Way pager, or SMS driver instance used by a Messaging Server process. For example, for an access point based on e-mail delivery, you create access points that correspond to the values entered for the *account names* parameter of the e-mail driver instance; if you configured an e-mail driver instance with the account name of *info* (and defined the `server.incoming.emails` parameter as *info@company.com*), you would create an access point for the account by entering info as the access point name, selecting e-mail as the delivery type and then entering *info@company.com* as the address. See also Section 3.10.3.1 and Section 3.6.2.2.2.

In addition to the basic information for the access point, the *Add Access Point* page enables you to create the following types of access points by using the *Allowed to Access All Applications* option:

- **Site access point** - An address that enables access to all the asynchronous applications. Select the *Allowed to Access All Applications* option to create a site access point.
- **Application category access point** - An address associated with one or more application categories. Content Managers associate these access points with an **application link category**. You create this type of access point by clearing (not selecting) the *Allowed to Access All Applications* option.

Figure 3–23 The Add Access Point Page


ORACLE ENTERPRISE MANAGER 10g
Application Server Control for Collaboration Suite
Logs
Topology
Prefere
Farm > Application Server: OCS10GM6BMid.omcollab03.us.oracle.com >
Wireless > Access Points >
Cancel
OK

Add Access Point

* Name	info
* Delivery Type	Email
* Address	info@company.com

For IM address, use the format of <network>|<User ID>. (Example: jabber|foo@jabber.org, yahoo|foo, msn|foo@msn.com, aim|foo, and icq|12345)

☐ Dedicated for Actionable Message Reply

☑ Allowed to Access All Applications

If allowed to access all applications, then this access point cannot be associated with any application categories.

See the *Oracle Application Server Wireless Developer's Guide* for more information on **Premium SMS**, **Reverse Charge SMS** and **actionable message**.

3.10.2.3 Configuring the Async Listener

The *Async Listener Configuration* page (Figure 3–21) enables you to configure the system settings for Async Listener, including the number of working threads, command format, application help, default application **short name**, and actionable message reply. Table 3–16 describes these and other configuration parameters for the Async Listener.

See the Oracle Application Server Wireless Developer's Guide for system configuration parameters for the Async Listener and for configuration parameters for actionable messages.

Table 3–16 Aysnc Listener Configuration Parameters

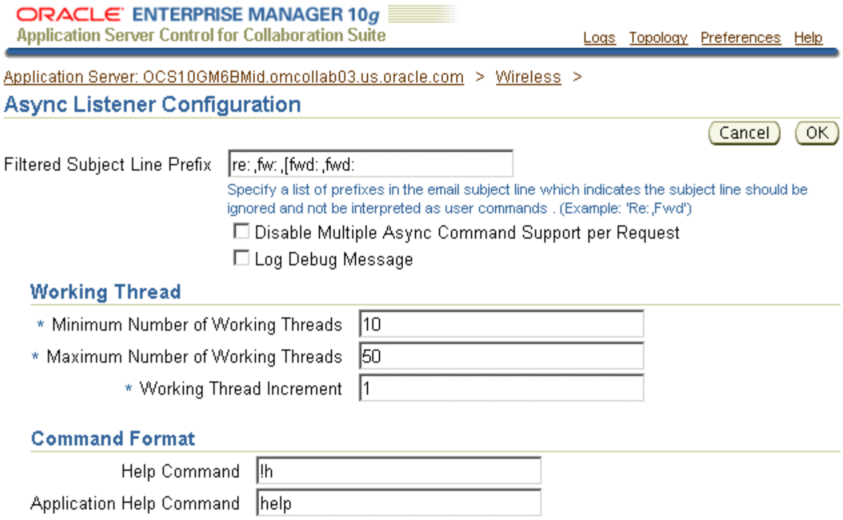
Parameter Name	Default Value(s)	Comments
Filtered Subject Line Prefix	re:,fw:,[fwd:,fwd:	Enter a list of prefixes for the e-mail subject line, which indicate that the message subject lines that start with these prefixes should be ignored and not be interpreted as user commands.
Disable Multiple Async Command Support per Request	N/A (You select a check box to enable this option.)	Selecting this option prohibits the Async Listener from sending back contents to users issuing commands to different applications in a single request. For example, a user issues a request to an access point configured to access all of the Async applications. These applications, in turn, request content from a stock application, a weather application, and a horoscope application. With this option set, the Async Listener only replies to the first command in the user's request string, thereby prohibiting the user from paying a single premium for content drawn from several applications.
Log Debug Message	N/A (You select a check box to enable this option.)	Selecting this option enables the Verbose logging mode.
Minimum Number of Working Threads	10	The number of working threads at the time Async Listener starts. The default value is 10 threads. Increase this number to accommodate a higher request rate.
Maximum Number of Working Threads	50	The maximum number of working threads used by the Async Listener.
Working Thread Increment	1	The number of threads to be added when there are no idle working threads available to serve requests. The thread increment stops once the maximum number set for the *Maximum Number of Working Threads* parameter is reached.
Help Command	!H	Provides general help on command usage.
Application Help Command	Help	Provides application-level help.
Escape command	!E	Clears current form state.
Stop command	!S	Marks the end of a command sequence.
Login command	!L	Enables user to sign on to the system with the user name and password.
Logoff command	!O	Signs off a user session.
Command Line Delimiter	;	The command separator used for a request with multiple commands.
Command Prefix	.	A symbol indicating that the text immediately after the symbol is an asynchronous application short name instead of a parameter value. This is useful when a user wants to escape out of a form state without having to use 'Escape command'. For example, the command *.stk orcl* with the period (.) as the command prefix.
Help Header	Usage -	The header of the applications help result. The *Help Header* and *Help Footer* values enclose the the application help information.

Table 3–16 (Cont.) Aysnc Listener Configuration Parameters

Parameter Name	Default Value(s)	Comments
Help Footer	N/A	The footer of the applications help result. The *Help Header* and *Help Footer* values enclose the the application help information.
Short Name for Default Application	Help	Enter the short name of an application. If the user request does not specify which application to invoke (or if the user requests an application that does not exist), then the Async Listener invokes this default application instead. The Async Listener invokes Help if no default application has been set.
Short Name for Replying to an Actionable Message	AM	The site-wide, unique name to identify the reply for an **actionable message**. The default is *AM*. No Async application can use the name specified in this field.
Maximum Active Transaction Number per Device	10	The number of transactions stored on the server for each user device.
Expiration Time for Non-Active Transactions (days)	20	The number of days that OracleAS Wireless stores a transaction if it has not been updated.

Note: The short name for replying to an actionable message must be unique among all the short name for asynchronous application links.

Figure 3–24 The Async Listener Configuration Page


ORACLE ENTERPRISE MANAGER 10g
Application Server Control for Collaboration Suite
Logs
Topology
Preferences
Help
Application Server: OCS10GM6BMid.omcollab03.us.oracle.com > Wireless >
Cancel
OK

Async Listener Configuration

Filtered Subject Line Prefix: re: ,fw: ,[fwd: ,fwd:

Specify a list of prefixes in the email subject line which indicates the subject line should be ignored and not be interpreted as user commands . (Example: 'Re:,Fwd')

□ Disable Multiple Async Command Support per Request

□ Log Debug Message

Working Thread

* Minimum Number of Working Threads: 10

* Maximum Number of Working Threads: 50

* Working Thread Increment: 1

Command Format

Help Command: !h

Application Help Command: help

3.10.2.4 Configuring the Async Listener as a Client of the Messaging Server

Because the Async Listener is a client of the Messaging Server, you must define the *Messaging Server Client* component. When configuring this component, you can add, delete or update the hooks used before or after sending a message (the pre-send and post-send hooks) or those used before or after receiving a message. Table 3–17 describes the parameters of the *Messaging Server Client* component of the Async Listener.

Table 3–17 Parameters of the Messaging Server Client

Parameter	Value
Thread Pool Size	The total number of threads created by the transport for this client. The transport uses these threads to retrieve received messages and status reports for this client. The transport ignores this setting if the client neither receives status reports nor has any registered end-points at which to receive messages.
Number of Queues	The number of queues. The transport creates this value only if this client receives status reports or messages. The transport supports only one queue per client; the transport creates only one queue per client even if you specify more than one queue per client. The number set at the site-level configuration is the default value if you do nor specify any value here. The transport ignores this setting if the client neither receives status reports nor has any registered end-points at which to receive messages.
Recipient Chunk Size	The number of recipients that receive messages in one send call by the client. If the number of recipients is too big, then the transport may send recipients messages on a chunk-by-chunk basis. In such cases, some may receive messages while the transport processes other recipients. As a result, some recipients get messages earlier than others. Sending messages chunk-by-chunk can improve performance. The chunk size cannot be more than 500; the transport uses a 500 chunk size even if the chunk size is set at greater than 500.
Carrier Finder Hook Class Name	OracleAS Wireless uses this hook to find the carrier name from a phone number. The carrier name is then used by the driver finder to find a proper driver to send a message to this phone number. Use this hook for situations where there are several carrier-specific drivers, as using a carrier's driver with a phone number of that carrier improves performance. If you do not specify the carrier finder hook class name at the node level, then OracleAS Wireless uses the one set at the site level. If you do not specify the carrier finder hook class name at the site level, then the driver finder cannot find an appropriate driver because it does not have the carrier information. If you do not specify the carrier finder driver hook class at either the site or node level, then OracleAS Wireless uses the transport's default driver finder.
Driver Finder Hook Class Name	The name of the hook that the transport uses to find an appropriate driver to send a message to a given destination. The driver finder hook uses such criteria as delivery type, cost, or speed to assign a driver. If you do not specify the driver finder hook class name at the node level, then OracleAS Wireless uses the driver finder hook specified at the server-level configuration.
■ Pre-Send Hook ■ Post-Send Hook ■ Pre-Receive Hook ■ Post-Receive Hook	These hooks can be called before or after sending a message (the pre-send and post-send hooks) or before or after receiving a message (the pre-receive and post-receive hooks). These hooks, which are in the same category, are called in the sequence in which they are specified. You can use these hooks to enable special client functions, such as checking or filtering, rather than having to implement an application on top of the transport.

3.10.3 Configuring Messaging

From the *Messaging* section (Figure 3–25), you can configure and manage the following components described in Table 3–18.

Table 3–18 Messaging Components

Component Name	Task
Drivers	Configuring the Messaging Server Drivers

Table 3–18 (Cont.) Messaging Components

Component Name	Task
Messaging Server Configuration	Setting the Default Configuration for the Messaging Server
XMS Configuration	Configuring the XMS Runtime

Figure 3–25 The Messaging Component

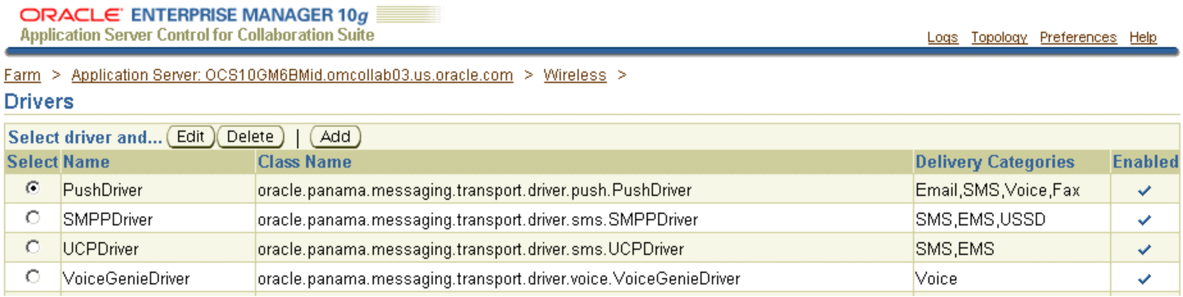
Messaging

Drivers
Messaging Server Configuration
XMS Configuration

3.10.3.1 Configuring the Messaging Server Drivers

The *Drivers* page (Figure 3–26), invoked by clicking *Drivers* enables you to define, edit, and delete a Messaging Server driver.

Figure 3–26 The Drivers Page



ORACLE ENTERPRISE MANAGER 10g
Application Server Control for Collaboration Suite
Logs Topology Preferences Help

Farm > Application Server: OCS10GM6BMid.omcollab03.us.oracle.com > Wireless >

Drivers

Select driver and... Edit Delete | Add

Select	Name	Class Name	Delivery Categories	Enabled
●	PushDriver	oracle.panama.messaging.transport.driver.push.PushDriver	Email,SMS,Voice,Fax	✓
○	SMPPDriver	oracle.panama.messaging.transport.driver.sms.SMPPDriver	SMS,EMS,USSD	✓
○	UCPDriver	oracle.panama.messaging.transport.driver.sms.UCPDriver	SMS,EMS	✓
○	VoiceGenieDriver	oracle.panama.messaging.transport.driver.voice.VoiceGenieDriver	Voice	✓

Table 3–19 lists the parameters of the Messaging Server drivers. Out of the box, OracleAS Wireless provides 15 seeded drivers, which support all of the delivery categories (SMS, e-mail, voice and fax). Each driver has a different set of class parameters. By default, all of these drivers are enabled. To improve performance, you can disable unneeded drivers.

3.10.3.1.1 Editing PushDriver PushDriver ships configured to connect to the Oracle-hosted online push service (`http://messenger.oracle.com/xms/webservices`) and supports limited delivery (1000 units) of messages through the SMS, voice, e-mail and fax delivery categories. When needed, you can remove delivery categories from PushDriver when you use a different driver and then create driver instance for that driver. For example, if you create an e-mail driver instance from the seeded-mail driver (EmailDriver), you must first edit PushDriver to remove its e-mail channel, so that all of the received e-mail messages are routed through the instance of EmailDriver.

See the *Oracle Application Server Wireless Developer's Guide* for information on drivers.

Table 3–19 Driver Parameters

Parameter	Description
Name	The name of the driver, such as EmailDriver. This is required parameter.
Class Name	The class name (with the full package name) that implements the driver.
Delivery Categories	The delivery category (or categories) of this driver, such as SMS, voice, or e-mail.
Enabled	Indicates that the driver has been enabled.

From this page, you can delete, edit, or create a messaging server drivers for the site. To create a new Messaging Server driver, you first click *Add Driver* and then define parameters listed in Table 3–20 in the *Add Driver* page.

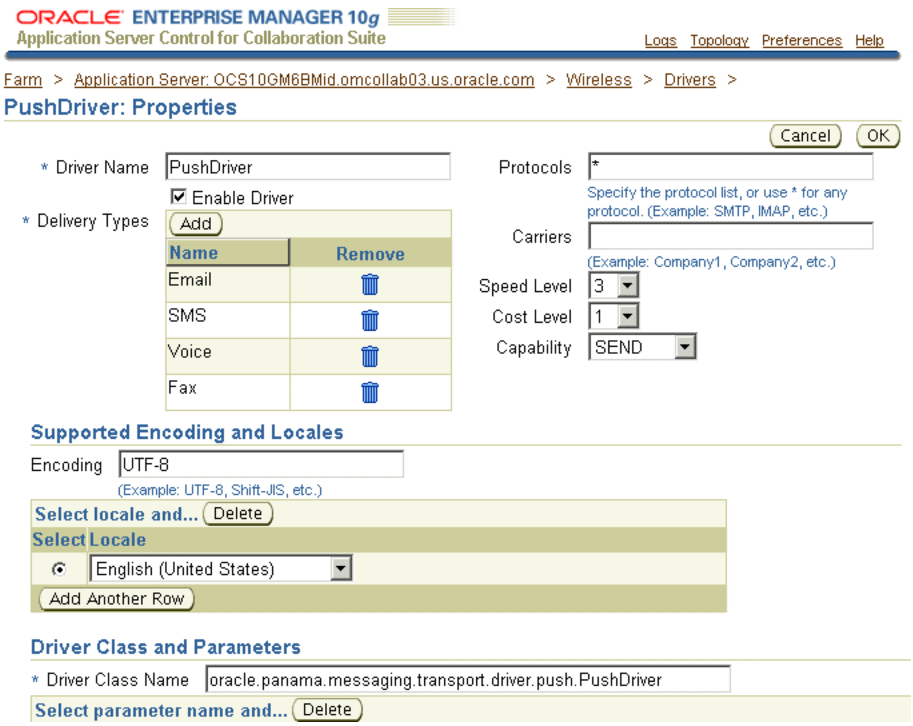
Table 3–20 Messaging Server Driver Parameters

Attribute	Description
Driver Name	A unique name for this driver. This is a required field.
Delivery Types	To designate how the Messaging Server driver delivers messages, select one or a combination of the delivery types: SMS, EMS, MMS, USSD, voice, e-mail, fax, WAP-Push, Two Way Pager, One Way Pager, or IM by clicking the **Add** button and then selecting the appropriate delivery types from the *Add Delivery Type* page that appears. From this page, you can select, remove or create a new delivery type for the driver should it require a delivery type other those provided out of the box. To create a new delivery type, enter a name for the delivery type and then click **Finish**. You must add at least one delivery type to a Messaging Server driver.
Enabled	Selecting this flag enables the Messaging Server to use this driver.
Protocols	A comma-separated list of protocols. Enter an asterisk (*) for any protocol.
Carriers	The comma-separated list of carriers.
Speed Level	The speed level of the driver. It can be from 0 to 10.
Cost Level	The cost level of the driver. It can be from 0 to 10.
Capability	This option sets the driver's ability to send or receive messages. The values can be *SEND*, *RECEIVE* or *BOTH* (which enables the driver to both send and receive messages). If you set the capability to *SEND*, then the driver has sending capability only (even if the `init()` method of a driver instance returns both sending and receiving capability). In addition, the driver instance will not be used to send any messages routed to this driver. Messages are still routed to this driver since the routing is based on OracleAS Wireless site-level driver configurations rather than on information from driver instances. Even if there are no instances of this driver, messages continue to be routed to this driver unless the driver has been disabled, or the driver does not match criteria specified by sending applications.
Supported Encoding	The supported encoding of this driver, such as UTF-8.

Table 3–20 (Cont.) Messaging Server Driver Parameters

Attribute	Description
Supported Locales	The supported locale list of this driver. You can add, remove or update the locale list.
Driver Class Name	The class name (with the full package name) that implements the driver, such as `Oracle.panama.messaging.transport.driver.email.EmailDriver`. *Driver Class Name* should be the full class name including the package. OracleAS Wireless does not check if such a class exists or if it is in the classpath, so you must therefore ensure that it is. Generally, it will be in the classpath if you put the class in `$ORACLE_HOME/wireless/server/classes`. All of the related classes should be put there except for `.jar` and `.zip` files, which should be put in `$ORACLE_HOME/wireless/lib`. This is a required parameter.
Driver Parameters	The driver class parameters. You can add, remove or update the parameters. Each parameter has multiple attributes, including: ■ Name -- The parameter name used by the driver class ■ Description -- The parameter description, such as the meaning of the parameter. ■ Mandatory -- Setting this flag marks the parameter as mandatory; not setting the flag marks the parameter as optional. ■ Default Value -- The default parameter value.

Figure 3–27 Editing the Driver Properties



ORACLE ENTERPRISE MANAGER 10*g*
Application Server Control for Collaboration Suite

Logs Topology Preferences Help

Farm > Application Server: OCS10GM6BMid.omcollab03.us.oracle.com > Wireless > Drivers >

PushDriver: Properties

Cancel OK

* Driver Name: PushDriver

☑ Enable Driver

* Delivery Types: Add

Name	Remove
Email	🗑
SMS	🗑
Voice	🗑
Fax	🗑

Protocols: *

Specify the protocol list, or use * for any protocol. (Example: SMTP, IMAP, etc.)

Carriers:

(Example: Company1, Company2, etc.)

Speed Level: 3

Cost Level: 1

Capability: SEND

Supported Encoding and Locales

Encoding: UTF-8

(Example: UTF-8, Shift-JIS, etc.)

Select locale and... Delete

Select	Locale
●	English (United States)

Add Another Row

Driver Class and Parameters

* Driver Class Name: oracle.panama.messaging.transport.driver.push.PushDriver

Select parameter name and... Delete

For information on configuring the SMPP driver, see Appendix 16.5.2 and Appendix 16.5.3.

3.10.3.1.2 Load Balancing by Driver Instances All of the voice driver instances inside a Messaging Server can balance load; if a driver instance ceases to function, then other instances take over and route messages to other voice gateways. Meanwhile, the instance checks to see if it can function again. The instance works again once the problem that prevented it from functioning is solved.

3.10.3.1.3 Configuring the VoiceGenie Driver Table 3–21 describes the voice flow-related parameters of the VoiceGenie driver (VoiceGenieDriver), one of the pre-built drivers that ships with OracleAS Wireless. (To view or edit these or other parameters, select VoiceGenieDriver from the *Drivers* page and then click **Edit**.)

Table 3–21 Voice Flow-Related Parameters of VoiceGenieDriver

Parameter	Description
`voice.response.seconds`	The interval, in seconds, after a call hangs up if no one answers. The default value is 60 seconds.This parameter applies to both the simple voice flow and the complex (fully tracking) voice flow.
`voice.max.queue.size`	The maximum number of messages that the driver can hold in memory. The default is 800 messages.This attribute applies to both the simple and the complex (fully tracking) voice flows.
`voice.delay.seconds`	The delay, in seconds, that the driver waits before checking if the voice gateway is up or down. The default is 40 seconds. This attribute applies to both the simple and the complex (fully tracking) voice flows.
`voice.calling.threads`	The number of actual calling threads. The default is 5. This attribute applies to both the simple and the complex (fully tracking) voice flows.
`voice.max.retries`	The maximum number of retry attempts when the phone or server is busy. There are eight retry attempts by default. This attribute applies to both the simple and the complex (fully tracking) voice flows.
`voice.flow`	Indicates the voice flow for text messages. Enter 1 for the regular text flow; enter 2 for the complex (fully tracking) voice flow. The default is 1 (the simple voice flow).
`voice.prompt.alert-notification-1`	Part 1 of the prompt alert notification. The default text is "This is an alert notification from Oracle Mobile. If you are...." This attribute applies only to the complex (fully tracking) voice flow (2).
`voice.prompt.alert-notification-2`	Part 2 of the prompt alert notification. The default text is "...say ’yes’ or press 1. Otherwise, say ’no’ or press 2. To repeat these options, say ’repeat’ or press 3." This attribute applies only to the complex (fully tracking) voice flow (2).
`voice.prompt.repeat`	The prompt for the user to repeat. The default is "I’m sorry, I didn’t understand. Please repeat your response." This attribute applies only to the complex (fully tracking) voice flow (2).
`voice.prompt.to-confirm`	The prompt for the user to confirm his (or her) entries. The default text is "To confirm receipt of this alert notification, say ’confirmed’ or press 1. If you would like to repeat the notification again, say ’repeat’ or press 3. This attribute applies only to the complex (fully tracking) voice flow (2).
`voice.prompt.good-bye1`	The reply to the user confirmation. The default text is "Thank you. Good-bye." This attribute applies only to the complex (fully tracking) voice flow (2).
`voice.prompt.good-bye2`	The reply if the user does not respond to voice.prompt.to-confirm. The default text is "I was unable to receive your response. Good-bye." This attribute applies only to the complex (fully tracking) voice flow (2).
`voice.prompt.alt-name`	The prompt for an alternative name. The default text is "...associated with, or employed by, Oracle." This attribute applies only to the complex (fully tracking) voice flow (2).

You can configure VoiceGenieDriver to support a either simple voice flow or a fully tracking, complex confirmed message delivery voice flow, which both prompts user inputs and provides responses to various user inputs. VoiceGenieDriver can only use only one type of voice flow at a time.

The Simple Voice Flow

By default, the VoiceGenie driver uses the simple voice flow. When the VoiceGenie driver uses the simple voice flow, a user hears the following upon receipt of a text message:

You have received a message. Sender: <sender>. Subject: <subject>. Message: <body>. You may say 'repeat' to hear the message again or hang up.

The Complex Voice Flow

When you configure the VoiceGenie driver to use the complex (fully tracking) voice flow, a user first hears the following upon the receipt of a text message:

```
<voice.prompt.alert-notification-1> <voice.prompt.alt-name>
<voice.promt.alert-notification-2>
```

The default script for these parameters is:

This is an alert notification from Oracle Mobile. If you are associated with, or employed by Oracle. say 'yes' or press 1. Otherwise, say 'no' or press 2. To repeat these options, say 'repeat' or press 3.

If the user does not reply, says "repeat", or presses 3, the VoiceGenie driver repeats. After it repeats three times, it continues with

```
<voice.prompt.good-bye2>
```

The default script is

I was unable to receive your response. Good-bye.

If the user says something other than "yes", "no", "repeat", or "1", "2", "3", the driver responds with

```
<voice.prompt.repeat>
```

The default script is

I'm sorry, I didn't understand. Please repeat your response.

If the user says "no" or presses 2, the VoiceGenie driver provides the response:

```
<voice.prompt.good-bye1>
```

The default script is

Thank you. Good-bye.

If the user says "yes" or presses 1, the voice flow replies

Subject: <subject>. Message: <body>. To confirm receipt of this alert notification, say 'confirmed' or press '1'. If you would like to repeat the notification again, say 'repeat' or press '3'.

The voice flow repeats three times if the user does not reply (or says something other than "repeat"), or does not press 3. After three times, the voice flow continues with

```
<voice.prompt.good-bye2>
```

By default, this script is

I was unable to receive your reponse. Good-bye.

If the user says, "confirmed" or presses 1, then the driver responds

Thank you. Good-bye.

Note: The Messaging Server generates a proper status report for each response if a status listener is registered. All status reports, which notify the sending applications of the status of the sent messages, are saved to the database even if no status listener has been registered.

3.10.3.2 Setting the Default Configuration for the Messaging Server

Clicking *Messaging Server Configuration* invokes the *Messaging Server Configuration* page, which enables you set the default configuration for the messaging server. Table 3–22 describes the Messaging Server configuration parameters.

Table 3–22 Messaging Server Configuration Parameters

Parameter	Description
GSM Smart Message Encoder Class Name	Enter the class name for the hook that encodes the GSM smart message (such as ring tone, graphics, WAP setting, and e-mail setting) for SMS.
Default Number of Sending Threads	The default number of sending threads for each driver instance. This value is used if the number of sending threads is not specified for a driver instance which has *SEND/BOTH* capability. See also Section 3.10.3.2.1.
Default Number of Receiving Threads	The default number of receiving threads for each driver instance. This value is used if the number of receiving threads is not specified for a driver instance which has *RECEIVE/BOTH* capability.
Send Retry Times	The maximum number of times that the Messaging Server retries sending a message after a failed send attempt. If the number of retries is met and the message is still not sent, then the message is moved to an exception queue. At this point, Messaging Server does not try to send the message again.
Send Retry Delay (second)	This interval, in seconds, sets the amount of time to wait before attempting to resend a message (after a failed attempt to send a message).

3.10.3.2.1 Determining the Number of Active Messaging Server Connections to the Database

The number of database connections that a Messaging Server process opens depends upon the configuration of its driver instance. If you configure the number of sending threads incorrectly (that is, the number of sending threads for the Messaging Server driver instance is greater than those configured for the site-level driver on which it is based), then the number of open, concurrent database connections may exceed those supported by the database.

You can configure the Messaging Server drivers to either send or receive, or to both send and receive messages. While the Messaging Server does not have a dedicated database connection for the receiving threads of a Messaging driver instance, it does have dedicated database connections for each sending thread of a Messaging Server driver instance. For example, if you configure a Messaging Server driver instance to have one sending thread, then the Messaging Server opens a single corresponding database connection for a driver supporting only one delivery type. The Messaging Server opens more database connections for driver instances that support multiple delivery types. In addition to the database connections opened by the Messaging Server, there is also a dedicated connection opened by the SMAgent (the panama server management agent).

Because the Messaging Server creates the database connections on demand, the total active database connections is calculated using the following formula:

$$1 + \sum_i (x_i * n_i)$$

In this formula, *i* is the index of a driver instance, *xi* is the number of sending threads of driver *i*, and is the default number of different delivery types supported by this driver instance which is configured at the site level.

For example, a Messaging Server process with:

- An e-mail driver instance that supports only e-mail and has three sending threads;
- A push driver instance that supports SMS and Voice and has one sending thread

Would have six active database connections: the thread opened by the SMAgent (1) + the number of push driver sending threads (1) * the push driver's delivery types (2) + the number of threads for the e-mail driver (3) * the number of sending threads (1).

3.10.3.3 Configuring the XMS Runtime

The *XMS Configuration* page enables you to configure the settings for XMS Runtime and enable the XMS Center (XMSC, described in Section 3.10.3.3.1), which adapts the content of a message to fit a given device. In addition, this page enables you to prioritize the device types for XMS message delivery.

Table 3–23 lists parameters that you define to set the XMS runtime.

Table 3–23 XMS Runtime Parameters

Parameter	Value
Server ID	The name of the XMS server, which is pre-pended to every message ID. This is an alpha-numeric value, which can range from one to four characters in length. This is an optional parameter.
Interval to Cleanup Processed Records (hours)	The frequency in which the database purges processed failover data. The default value is 48 hours.
Maximum Days to Keep Request in Failover Table	The maximum lifetime for a failover record in the database. All failover records, whether they have been processed or are still pending, are deleted after this period. The default value is 30 days.
Maximum Levels of Failover Supported	The maximum number of failover address-delivery types channels allowed per recipient. The portion which exceeds the limit will be truncated and lost. The default value is 5.
Number of Status Receiving Threads	The number of threads used to retrieve the sending status from the Messaging Server. Increasing the number of transport receiving threads decreases the backlogs in the transport status queue, thereby improving the performance of the status callback. The default value is four threads; the maximum number of status receiving threads is 30.
Number of Failover Processing Threads	The default value for the number of failover processing threads is 2 threads, which is the minimum value for this parameter. The maximum number of failover receiving threads is 30. Increasing the number of failover threads improves performance and results in shorter delays.
Number of HTTP Status Callback Threads	The number of threads which send the HTTP delivery status update notifications to XMS clients. The default value is 4 threads, which is the minimum value for this parameter. The maximum number of threads is 30. Increasing the number of threads improves performance and results in shorter delays for status notification delivery.

Table 3–23 (Cont.) XMS Runtime Parameters

Parameter	Value
Status Message Lifetime (Hours).	The maximum lifetime for a status message. The message is discarded after its lifetime expires even if it has not been delivered to an XMS client. The default value for the lifetime of a status message is 24 hours; the minimum value for this parameter is 1 hour. The maximum value for this parameter is 120 hours.
Enable HTTP Callback Retry	Select **Yes** to enable the XMS server to send delivery status update notifications to the sender of a message through HTTP. Selecting **No** to disable this feature causes the status message to be discarded immediately if it cannot be delivered to the sender of the message. If you select **Yes**, you must also set the time, in seconds, for the *HTTP Callback Retry*.
HTTP Callback Retry (seconds)	This parameter defines the retry frequency when the XMS server cannot send a delivery status update notification to the sender when the *HTTP Callback Retry* is enabled. The default value is 5 seconds, which is the minimum value for this parameter. Increase this interval (the maximum value is 60 seconds) to improve performance by decreasing the retry frequency.
Status Message Formatter Hook Class	Enter the full path of formatter hook Java class. The Status Message Formatter formats the human-readable delivery status description text so that is more "machine-readable". This is an optional parameter.
Billing Hook Class Name	The full path of the billing hook Java class This is an optional parameter.

3.10.3.3.1 Configuring the XMS Center The XMS Message Center (XMSC) supports MMS Center functionality out of the box, so that a device with an MMS browser can receive notification messages and retrieve messages stored on the OracleAS Wireless Server through HTTP. It also supports MO (mobile-originated) messages to another phone, message storage and notifications for delivery channels other than MMS. To configure the XMSC, define the following two parameters:

- **Enable XMSC** — Selecting this option enables the XMSC. By default, XMSC is enabled (set to *true*). When you enable the XMSC, notification messages are sent out automatically when you send a message through an MMS notification message.
- **Message Life Time —** The maximum amount of time that a message can be stored on the server and available for retrieval by users. The default period is 7 days.

3.10.3.3.2 Configuring the Delivery Channels XMS supports implicit device, or user addressing, by specifying the OracleAS Wireless user name. XMS selects the best device for the user to receive messages, based on such factors as messaging content, application hints, and user preferences. If OracleAS Wireless cannot send a message to one device, then XMS fails over to the next device in selection order and transforms the content for that device.

You define the values the *Delivery Channel Settings* section by specifying the priority (or failover) of the XMS message delivery types and by adding the appropriate reply addresses for the delivery types.

3.10.4 Configuring the Notification Engine

Table 3–24 describes the components of the Notification Engine and their related configuration tasks.

Table 3–24 Notification Engine Components

Component	Task
Notification System	Configuring the Notification System Reply Address
Messaging Server Client	Configuring the Notification Engine as a Client of the Messaging Server

Figure 3–28 The Notification Engine Component

3.10.4.1 Configuring the Notification System Reply Address

You can configure reply addresses of notifications for:

- E-mail
- SMS
- Pager
- Voice
- WAP Push

You can also configure the following runtime parameters:

- *Number of Location Event Listener Threads* — The number of threads to start for each notification process for listening to incoming location events. The default is 1.
- *Location Condition Response Delay (seconds)* — The approximate response delay for location condition processing. The default is 600.
- *Message Manager Thread Pool Size* — This is the number of worker threads in a message manager. The default value is 5.
- *Event Handler Block Time (seconds)* — The interval, in seconds, that the event handler stops adding new event jobs to the message manager job queue when it becomes overloaded. The default value is 60000.
- *Event Handler Block Timeout (seconds)* — The block timeout (in seconds) after which the event handler checks the worker threads if the manager job queue is still overloaded. The default value is 600000.
- *OC4J Check Interval* — The interval in which to check if the OC4J instance is running. Note: A running OC4J instance is required to start the Notification Engine. The default value is 120000.
- *OC4J Wait Timeout (milliseconds)* — The timeout for the Notification Engine to wait for the OC4J instance to start. The default value is 1200000.
- *Notification Check Interval (milliseconds)* — The interval for the refire thread to calculate the sending time for the next set of time-based notifications. The default value is 3600000.
- *Stop Request Check Interval (milliseconds)* — The interval for the refire thread to check if the notification engine has received a stop request so that it can gracefully exit on such a stop request. The default value is 900000.

- *Maximum Number of Exceptions for Spin* — The maximum number of consequent exceptions that can be thrown before the worker thread enters the spin mode. The default value is 20.
- *Spin Block Time (milliseconds)* — The time to sleep when the worker thread enters the spin mode. The default value is 60000.

3.10.4.2 Configuring the Notification Engine as a Client of the Messaging Server

Because the Notification Engine is a client of the Messaging Server, you must configure the Messaging Server Client component. For more information, refer to Section 3.10.2.4.

3.10.5 Configuring the Location-Related Components

The location-related configuration includes the following:

- **Location Management** -- For mobile positioning configuration, mobile positioning provider information and configuration, and mobile ID names.
- **Location Services** -- For configuration options relating to geocoding, routing, mapping, traffic, and business directory services.
- **Location Event Server** -- For options relating to the location event server.
- **Location Mark Address Format** -- For specifying location mark address fields.

> **Note:** For more information on the location-related components of OracleAS Wireless, refer to the *Oracle Application Server Wireless Developer's Guide*

Location Mark Address Format

This page enables you to configure the format of location mark address. To do this, you select all of the attributes that you want to display for a location mark address. This configuration is used in the Customization Portal.

- Company Name
- Address Line 1
- Address Line 2
- Address Last Line
- Block
- City
- State
- Postal Code
- Postal Code Extension
- County
- Country

3.10.6 Configuring the Provisioning Server

The OracleAS Wireless Provisioning Server enables application providers to create and publish applications as well as serve content to the end-users when they download a selected application. The download protocol used depends upon the application type (such as a J2ME MIDlet or a ring tone) and the line provisioning protocol. The *Provisioning Server* page enables you to define the class names for the pre-download hook, the post-download hook, and the deliverable content event listener. You can also add new drivers or implement the customized functions of the existing drivers.

> **Note:** Oracle Application Server Wireless 10*g* Release 2 (10.1.2.02) supports only J2ME applications.

Hooks

The actual upload and download processes can be monitored using hooks, which customers implement. The hooks are initialized using a singleton pattern. The hook method is given the user information, the application information and the content information. The hook implementation must provide a method such as:

```
public static <hookclass> getInstance()
```

Pre-Download Hook Class Name: This hook is invoked just before the user downloads the application. The hooks are initialized using a singleton pattern. The return code of the hook determines if the download can proceed.

The interface to be implemented is:

```
oracle.panama.rt.hook.ProvisioningPreDownloadHook
```

Post Download Hook Class Name: This hook is invoked once just after the user downloads the application and once after the user's device notifies the server of the application download. The provider can embed the billing action in either of these two invocations as appropriate.

The interface that you implemented is:

```
oracle.panama.rt.hook.ProvisioningPostDownloadHook
```

Deliverable Content Event Listener Class Name: This hook is invoked during content upload, update or delete.

The interface that you implement is: `oracle.panama.rt.event.DeliverableCtntEventListener`

The driver implements the following interface: `oracle.wireless.me.provisioning.ProvisioningDriver`

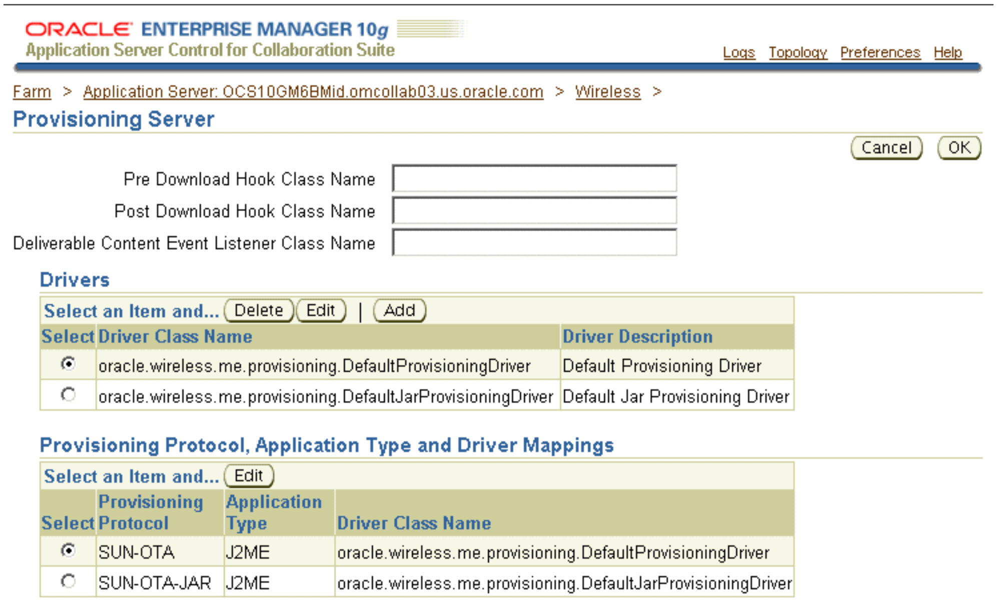
Drivers

You can add, delete or edit a driver. To add a driver, provide the driver class name, driver description and driver parameters, if any. Out of the box, OracleAS Wireless provides two driver implementations: the default provisioning driver (`oracle.wireless.me.provisioning.DefaultProvisioningDriver`) and the default JAR provisioning driver (`oracle.wireless.me.provisioning.DefaultJarProvisioningDriver)`These drivers are mapped to download J2ME MIDlets and JAR files, respectively.

Driver Mapping

You map the driver used for the appropriate application type and protocol configuration. Out of the box, the two seeded drivers support SUN-OTA, and SUN-OTA_JAR protocols for J2ME applications. You can select the driver classes, which are used for the two protocols.

Figure 3–29 The Provisioning Server Page


ORACLE ENTERPRISE MANAGER 10g
Application Server Control for Collaboration Suite
Logs
Topology
Preferences
Help
Cancel
OK
Delete
Edit
Add
Edit

Farm > Application Server: OCS10GM6BMid.omcollab03.us.oracle.com > Wireless >

Provisioning Server

Pre Download Hook Class Name

Post Download Hook Class Name

Deliverable Content Event Listener Class Name

Drivers

Select an Item and...

Select	Driver Class Name	Driver Description
●	oracle.wireless.me.provisioning.DefaultProvisioningDriver	Default Provisioning Driver
○	oracle.wireless.me.provisioning.DefaultJarProvisioningDriver	Default Jar Provisioning Driver

Provisioning Protocol, Application Type and Driver Mappings

Select an Item and...

Select	Provisioning Protocol	Application Type	Driver Class Name
●	SUN-OTA	J2ME	oracle.wireless.me.provisioning.DefaultProvisioningDriver
○	SUN-OTA-JAR	J2ME	oracle.wireless.me.provisioning.DefaultJarProvisioningDriver

3.11 Uploading and Downloading Repository Objects

The *Utilities* section contains the common administrator utilities.

Note: For the utilities to function:

- You must configure the *OracleAS Wireless Tools URL* correctly. If you use instance URLs, then you configure this URL using the *Instance URLs* page accessed from the *Home* page. For more information, see Section 3.5.1. If you use site URLs, then you configure the *OracleAS Wireless Tools URL* from the *HTTP, HTTPS Configuration* page accessed from the *Site Administration* tab. For more information, see Section 3.9.1.1.
- The OracleAS Wireless Tools must be running, because the actual functions are hosted there.

3.11.1 Repository Objects Download

The Repository Objects Download page, invoked by selecting the *Repository Objects Download* link from the *Utilities* section of the *Administration* page, enables you to download repository objects. You can specify the types of repository objects to download. For example, you can download only adapters.

In addition, you can download by OID, and you can download applications by folder, or by user. You can also download all objects by user.

You can only download repository objects to a local file.

To download repository objects:

1. Enter the location for the log files.
2. Enter the location of the logging activity. This is a server-side generated log file. For example, enter */temp/activity.log*.

3. Enter the location for logging errors. This is a server-side generated log file. For example, enter */temp/error.log*.

4. Specifying the *Objects for Download* by entering the filter expression for the name of the objects to be extracted. For example, enter \”*/home/master**\”. You can include wildcards, such as *[*%_]*.

> **Note:** This filter expression applies only to downloading specific types of objects, such as groups, or adapters. This filter does not work if the *Download All Objects* or *Download by Object ID*, *Download by Users*, or *Download by Folder* options.

5. Select from among the following options:

   - Download All Objects
   - Download All Adapters
   - Download All Devices
   - Download All Groups
   - Download All Location Marks
   - Download All Applications
   - Download All Transformers
   - Download All Users
   - Download All Master Notifications
   - Download All Notifications (deprecated)
   - Download All Data Feeders
   - Download All Topics (deprecated)
   - Download All Subscriptions
   - Download All Application Categories
   - Download All Application Category Access Points
   - Download All Application Access Points
   - Download by Object ID (OID). You must enter a range or comma-separated list of OIDs. Use a comma (,) to separate your entries.
   - Download Applications by Folder. For this option, you must enter the folder path or folder URL.
   - Download Applications by User Name. You must enter the user name. You cannot enter multiple user names.

6. Click *Download*. A Windows dialog appears.

In the Windows dialog, specify the local XML file for the downloaded objects. Clicking *Cancel* after Download stops the download operation.

Note: If you have not yet performed the Single-Sign-On (SSO) login, then OracleAS Wireless redirects you to the *SSO* page the first time you click the *Download* button. On the *SSO* page, you enter a valid Superuser's user name and password. OracleAS Wireless then prompts you with the download dialog to specify the file location. After that, you remain at the SSO login page. To return to the OracleAS Wireless download page, click the browser's *Back* button. The next time you click the *Download* button, you will not be redirected to the SSO page because you are already logged into SSO.

3.11.2 Repository Objects Upload

You can upload repository objects from a local file.

The upload function performs the following:

- Checks for the objects in the repository by logical unique name.
- Loads all dependencies.
- If the objects exist in the repository, then the uploading facility updates the objects.
- If the objects do not exist, then the uploading creates them.

After each object type is successfully loaded, the uploading facility performs a commit unless you specify a different commit frequency. The commit includes all referenced objects (dependencies).

OracleAS Wireless does not validate the XML file that you import into the repository using the upload facility. To avoid errors, work in an XML file that you have exported from the repository. This gives you a "known good" Repository XML framework for adding, removing, and modifying individual elements.

To upload repository objects:

1. Enter the name and location of the file you want to upload, or select it using the Browse function.
2. Enter the location of the logging activity. This is a server-side generated log file. For example, enter */temp/activity.log*. This is a required field.
3. Enter the location for logging errors. This is a server-side generated log file. For example, enter */temp/error.log*. This is a required field.
4. Enter the number of objects uploaded that triggers a commit. Entering *0* causes a commit after the utility has completed the upload.

Note: In integrated mode, you must be logged into SSO as a valid user with the Superuser's role before you can upload objects successfully

The Repository Objects Upload page, invoked by selecting the *Repository Objects Upload* link from the *Utilities* section, enables you to upload repository objects.

4

Managing Oracle Sensor Edge Services

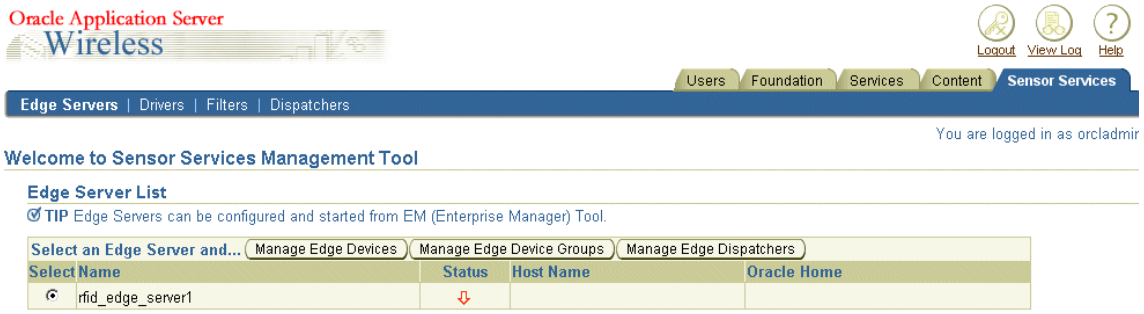
This chapter, through the following sections, describes how to use the Sensor Services tool.

4.1 Overview of the Sensor Services Management

The Oracle Sensor Edge Server enables enterprises to incorporate information from sensors into their I.T. infrastructure and business applications by receiving event data from sensor devices or applications and then normalizing this data by putting it in a common data format and stripping it of extraneous information using filters. The event data, which is now a normalized event message, is then sent to edge clients using a dispatcher. Depending on the configuration of the Oracle Sensor Edge Server's dispatcher, an Oracle Sensor Edge Server client receives event messages through database streams, JMS (Java Message Service), Web Services, or HTTP. The payload of the message is always an event.

The *Sensor Services* tab (Figure 4–1) enables you to manage how an Oracle Sensor Edge Server process receives, filters, and dispatches data.

Figure 4–1 The Edge Server Browsing Page



You are logged in as orcladmir

Welcome to Sensor Services Management Tool

Edge Server List

TIP Edge Servers can be configured and started from EM (Enterprise Manager) Tool.

Select an Edge Server and...

Select	Name	Status	Host Name	Oracle Home
●	rfid_edge_server1	⇩		

For more information on Oracle Sensor Edge Server processes, see Section 3.6.1, "Managing and Configuring the Web-Based Applications".

4.1.1 Overview of Events

The Event type represents a message sent from one component of the Oracle Sensor Edge Server to another. These event messages are specific to each type of driver or filter. The Event type is divided into the following sections:

- Header
- Type Info
- Payload

Header

The *Header* sections includes the routing headers and the message headers, which contain fields that designate the delivery of an event message. The routing fields include `<sourceName>` and `<correlationID>`. The message headers include the `<siteName>` and `<deviceName>` fields.

- `<sourceName>`

  This field identifies the originator of the event. This is an optional field, one set by the client.

- `<correlationId>`

  The client sets the value for this field, which is used for message responses to a particular client (such as checking if a device functions). Any message sent back by the client has the same ID. This is an optional field.

- `<siteName>`

  The site that originally generated the message.

- `<deviceName>`

  The name of the device or application that generates the event.

- `<time>`

  The date and time when the observation or message was created.

Type Info

The *Type Info* section contains the formatting information for the payload: the type and subtype of the event.

- `<type>`

  The number value that corresponds to the type of event. Table 4–1 describes the values of the `<type>` field. The *Oracle Application Server Wireless Developer's Guide* describes the values for the `<type>` field in further detail.

Table 4–1 Value Ranges for the <type> Field

Range	Message Type
0-99	System messages. The range of values includes: ■ 0 -- Unknown A value of 0 represents a bad event or a system internal event. ■ 1 -- Message Event A confirmation message, usually the return result code of a corresponding instruction.
100-199	Instructions or commands. The range of values includes: ■ 100 -- General Instructions General Instructions for controlling devices. ■ 101 -- **Radio Frequency Identification (RFID)** Instruction Instructions to RFID devices. ■ 102 -- Printer Instructions to printers. ■ 103 -- Lightstack
200-299	Observations. The range of values includes: ■ 200 -- RFID Observation The message is an RFID observation ■ 201 -- **Real Time Location System (RTLS)** ■ 202 -- Physical Contact ■ 203 -- Temperature ■ 204 -- Humidity ■ 205 -- Weight ■ 206 -- Tampering ■ 207 -- Audio ■ 208 -- Message Board
500-599	Custom messages

- `<subtype>`

  The number value for the subtype. For more information on Instruction Events, refer to the *Oracle Application Server Wireless Developer's Guide*

Payload

The *Payload* section, which is the event-specific section with the following fields:

- `<id>`

  The text value of this field identifies a **tag** (that is, a **read** or target) to an event instruction.

- `<data>`

The tag data. This is an optional field.

4.1.2 Managing Edge Server Objects

The Sensor Services tool includes the following tabs, which enable you to enhance or change the capabilities of the Oracle Sensor Edge Server processes by adding extensions (custom-built drivers, filters and dispatchers) to the repository and administer the Oracle Sensor Edge Server processes themselves.

- Edge Servers
- Drivers
- Filters
- Dispatchers

Edge Servers

The Edge Servers tab (Figure 4–1), which opens by default when you access the Sensor Services tool, includes a table that lists the Oracle Sensor Edge Server processes. (Users granted the System Manager role create and configure these processes as well as start and stop them.) From this screen, you can view the current status (running or stopped) of an Oracle Sensor Edge Server processes, the Oracle Sensor Edge Server group to which it belongs and the node information for the process, the Oracle Home and Oracle Host of the process. The *Oracle Home* and *Oracle Host* values displayed for a process enable you to locate the installation of the Oracle Sensor Edge Server and distinguish between multiple instances of the Oracle Sensor Edge Server running on the same Oracle Host. Multiple Oracle Sensor Edge Server instances can be installed inside different Oracle Homes, but can run on the same Oracle Host.

> **Note:** The Oracle Home and Host Name values only appear in the integrated mode.

For more information on configuring the Oracle Sensor Edge Server processes, see Section 3.6.1, "Managing and Configuring the Web-Based Applications".

From this table, you assign devices to (or delete devices from) a selected Oracle Sensor Edge Server process by clicking the **Manage Edge Devices** button. Using the functions accessed through this button, you can also edit the properties of a device, such as the filters (that is, the instances of a filter object) that it uses. Likewise, clicking the **Manage Edge Device Groups** button enables you to add device groups (the logical groupings of Oracle Sensor Edge Server devices), to a selected Oracle Sensor Edge Server process, manage the device membership of the group, manage the filter assigned to the group, or delete the device group from the Oracle Sensor Edge Server process. The **Manage Edge Dispatchers** button enables you to assign a single dispatcher to a selected Oracle Sensor Edge Server process.

> **Note:** If you modify the assignments of devices, device groups, or edge dispatchers of an Oracle Sensor Edge Server process, or modify the devices, device groups, or dispatchers themselves, then you must use the System Manager to stop and restart the Oracle Sensor Edge Server process to which these components belong, so that any changes can take effect. For more information on starting and stopping a standalone instance, see Section 3.6.2.1, "Creating a Standalone Process".

Drivers

From the *Drivers* tab, you can add the drivers used by the devices belonging to an Edge Server process by uploading an Extension Archive file (a JAR file containing all of the class files as well as other related files of the driver) to the repository. For information on creating and managing drivers, see Section 4.3. See the *Oracle Application Server Wireless Developer's Guide* for more information on creating the Extension Archive file. The drivers are not executed. For the Oracle Sensor Edge Server to use a driver, you must create an instance of that driver called a device.

Filters

Similarly, the *Filters* tab also enables you to add a filter by uploading an Extension Archive file (a JAR file containing all of the class files as well as other related files of the filter) to the repository. These filters, which strain out unneeded event data, become available to Edge Server devices when you assign a filter to a device. Once a filter is assigned to a device, a filter instance (an object of the filter) is created. For information on creating and managing filters Section 4.4. See the *Oracle Application Server Wireless Developer's Guide* for more information on creating the Extension Archive File.

Dispatchers

Using the *Dispatchers* tab, you add a dispatcher, an object that forwards event data from the Oracle Sensor Edge Server, by uploading an Extension Archive file (a JAR file containing all of the class files as well as other related files of the dispatcher) to the repository. For an Oracle Sensor Edge Server process to have dispatcher functions, you must assign a dispatcher to a process. When you assign a dispatcher to an Oracle Sensor Edge Server process, you create an edge dispatcher (an instance of a dispatcher). For more information on creating and managing dispatchers, see Section 4.5. See the *Oracle Application Server Wireless Developer's Guide* for more information on creating the Extension Archive File.

Note: The drivers, filters, or dispatchers are available to all Oracle Sensor Edge Server processes in the repository after you upload them to the repository.

4.2 Accessing Sensor Services Management

After you log in to the OracleAS Wireless Tools, you select the Sensor Services Tool by clicking the *Sensor Services* tab.

Note: You must be granted either the Super-user or Oracle Sensor Edge Server Manager roles to access the Sensor Services tool.

4.3 Managing Drivers

Drivers enable communication between a device and the Oracle Sensor Edge Server.

The driver browsing screen enables you to create, edit and delete drivers. The screen includes a list (Table 4–2) that organizes the drivers available for creating devices as follows:

Table 4–2 Elements of the Driver Browsing Screen

Element	Description
Name	The name of the driver.
Class Name	The name of the driver implementation class.
Version	The version of the driver implementation.
Description	A description of the driver.
Created Date	The date that the driver was created.

4.3.1 Adding a Driver

To add a driver, first click **Create**. The *Create New Driver* screen appears. Using the **Import** button, locate and then upload the driver's Extension Archive file to the repository. The Extension Archive file is a JAR file containing all of the class files and native binaries of the driver, as well as its properties files or static data files. In addition, the Extension Archive includes the Extension Archive Descriptor file, an XML file containing instructions for the Oracle Sensor Edge Server on loading and managing the driver.

Once the driver's Extension Archive file has been uploaded, the parameters specific to the uploaded file appear. (Unless the Extension Archive Descriptor file has been modified, the driver parameters are read-only.) Click **Finish** to complete the driver. The driver, which then appears in the browsing screen, can be used to create a device for the Oracle Sensor Edge Server process (described in Section 4.6).

Note: After you create a device, you must stop and restart the Oracle Sensor Edge Server process using the System Manager. For more information on starting and stopping a standalone instance, see Section 3.6.2.1, "Creating a Standalone Process".

For information on the packaging an Extension Archive and the Device Management API, see the *Oracle Application Server Wireless Developer's Guide*

4.3.2 Deleting a Driver

To delete a driver from the repository, first select a driver from the list and then click **Delete**. Deleting a driver also results in the deletion of the devices and filters using that driver.

4.3.3 Configuring the Pre-Seeded Drivers

OracleAS Wireless provides the following drivers out of the box:

- EdgeSimulator (Section 4.3.3.1)
  - Version: 10.1.2
- Alien (Section 4.3.3.2)
  - Model: 9RE-001
  - SDK Version: Alien SDK Version 2.1.0
- Intermec (Section 4.3.3.3)
  - Models: Penn Reader, Delaware Reader, PCMCIA Reader

  - SDK Version: COM API Version 2.0
- Patlite (Section 4.3.3.4)
  - Model: New PHE-3FB PC-Controlled Light
  - Protocol: PHE-3FB System Control Protocol

> **Note:** The Intermec and Lightstack Device Controllers are available for download from Oracle Technology Network (http://www.oracle.com/technology/).

4.3.3.1 Configuring the Edge Simulator

The Simulator (the EdgeSimulator driver) generates events to simulate a real device. In general, you use the EdgeSimulator driver to test configurations and deployment designs; however, you can also use it for internal functional testing to see how events are processed throughout the system. The EdgeSimulator driver acts the same as any driver, except that instead of connecting to a physical device to read events, it takes parameters from an input file (such as Example 4–3) as instructions on when to generate fake events. This begins as soon as the device starts (which starts when the Oracle Sensor Edge Server starts).

4.3.3.1.1 Configuration Steps

To configure the Simulator, enter the name of the input file, an XML file that tells the simulator how to generate the fake events using the following instructions:

- `<EventList>`

  The `<EventList>` element defines a loop. This element is also the main block that groups all of the other instructions together. `<EventList>` has one attribute, repeat, which must be present to control looping. The value for repeat must be a decimal number from 0 to `LONG_MAX`. To generate events only once, set the repeat attribute to 1. Setting repeat to n results in all instructions looping *n* times. Setting repeat to 0 disables the block and causes the parser to skip it.

  Example 4–1 illustrates the syntax for generating two events, pausing, generating two more events, and then looping 20 times:

Example 4–1 Defining a Loop

```
<EventList repeat='20'>
<Event> … </Event>
<Event> … </Event>
<EventInterval>…</ EventInterval>
<Event> … </Event>
<Event> … </Event>
</EventList>
```

  You can include any number of instructions inside the `<EventList>` element. The order in which they are defined is the order in which they are executed.

- `<EventInterval>`

  The `<EventInterval>` element instructs the Simulator to pause for a certain period of time before proceeding. This is usually used to throttle the data rate. A decimal number defines the time period, in milliseconds, to wait for before executing the next instruction. Section 4–2 illustrates how to instruct the Simulator to wait for half a second between each event and three seconds between loops:

Example 4–2 The <EventInterval> Element

```
<EventList repeat='20'>
   <Event> … </Event>
   <EventInterval>500</ EventInterval>
   <Event> … </Event>
   <EventInterval>500</ EventInterval>
   <Event> … </Event>
   <EventInterval>3000</ EventInterval>
</EventList>
```

- `<Event>`

  The `<Event>` element tells the Simulator to send an event. The child elements (described in Table 4–3) control the event's fields.

Table 4–3 Event Elements for the Simulator

Event Field	**Value**
`<type>`	The number value that corresponds to the type of event. Table 4–1 describes the values of the `<type>` field. The *Oracle Application Server Wireless Developer's Guide* describes the values for the `<type>` field in further detail.
`<subtype>`	The number value for the subtype. For example, the subtype value in Example 4–3 corresponds with a General Instruction Event, which is an event sent by application or a device to tell a specific device to perform an operation. In Example 4–3, the value of 1 turns on the device. Refer to the *Oracle Application Server Wireless Developer's Guide* for more information on Instruction Events.
`<id>`	The text value of this field identifies a tag (that is, a read or target) to an event instruction. In Example 4–3, one of the `<id>` values for a tag is 03ffff045679.
`<data>`	The tag data. This is an optional field.
`<deviceName>`	The name of the device or application that generates the event. The `<deviceName>` enables the Simulator to appear as if it is another device when generating events.

Example 4–3 illustrates an input file which includes two groups of events: the first one runs only once and the second runs 20 times.

Example 4–3 Simulator Input File

```
<EdgeEventSimulation>
      <EventList repeat='1'>
         <Event>
            <type>100</type>
            <subtype>1</subtype>
            <id>03ffff045679</id>
            <data>No Data</data>
            <deviceName>My Device</deviceName>
        </Event>
   <EventInterval>500</ EventInterval>
      <Event>
         <type>100</type>
            <subtype>1</subtype>
            <id>03ffff045680</id>
            <data>No Data</data>
            <deviceName>My Device</deviceName>
     </Event>
          <EventInterval>3000</ EventInterval>
   </EventList>

   <EventList repeat=’20’>
      <Event>
         <type>100</type>
         <subtype>1</subtype>
         <id>04ffff045679</id>
         <data>No Data</data>
         <deviceName>My Device</deviceName>
    </Event>
         <EventInterval>500</ EventInterval>
     <Event>
         <type>100</type>
          <subtype>1</subtype>
          <id>04ffff045680</id>
          <data>No Data</data>
          <deviceName>My Device</deviceName>
     </Event>
   </EventList>
</EdgeEventSimulation>
```

Although the format of the Event type is fixed, you can extend the Event type by mapping its fields to different meanings depending on the type of event.

4.3.3.2 Configuring AlienDevice Driver

The AlienDevice driver supports all of the Alien Technology RFID readers. The ALR series of readers has been tested for this release of the Oracle Sensor Edge Server. Configuring the AlienDevice driver includes:

- Finding the IP Address of the Alien RFID reader using the discoverer included in the RFID Gateway demo software provided by Alien Technology (Section 4.3.3.2.1).
- Connecting the Alien RFID reader to a Web browser (Section 4.3.3.2.2).
- Connecting the Alien RFID reader to the Oracle Edge Server by creating a device using the AlienDevice driver (Section 4.3.3.2.3).

4.3.3.2.1 Finding the IP Address of the Alien RFID Reader By default, the Alien RFID reader uses DHCP (Dynamic Host Configuration Protocol) to get its IP address upon connection to the network. To discover the IP address without a DCHP server, use the RFID Gateway Demo software, available as part of the Nanoscanner Development Kit Releases, as follows:

1. Install the RFID Gateway Demo software.
    1. Run the `setup.exe` (on the Alien Technology CD).
    2. Select **Install RFID Gateway Demo Software** and follow the installation instructions. A folder called *Alien RFID Gateway* appears in the *Programs* folder of the *Start* menu.
2. When the RFID Gateway Demo Software is installed, use it to discover the IP address as follows:

1. Start the Alien RFID software by selecting **Alien RFID Gateway** (located in the *Alien RFID Gateway* folder of the *Start* menu). Upon startup, the Alien RFID Gateway software scans the serial ports of the computer and the network and displays a list of the Alien RFID readers and their IP addresses in the window that appears. For example, a reader might be listed as *Alien RFID Reader@144.25.171.209*.
2. Get the IP address of the Alien RFID Reader from the list. Using this address, you can configure the Alien RFID reader to the Web browser.

4.3.3.2.2 Connecting the Alien RFID Reader to a Web Browser Because the Alien reader has a built-in Web Server, you can connect it to a Web browser by pointing the browser to the IP address of the Alien device, such as `http://144.25.171.209`. When prompted, enter a user name and password. Because the default settings should be correct, you can then create a device from the RFID reader, which is an instance of the device. For more information on creating devices, see Section 4.6.

4.3.3.2.3 Creating a Device for the Alien RFID Reader To connect the Alien RFID reader to the Oracle Edge server, create a device for the Alien RFID reader:

1. Enter the name of the device. This is a mandatory parameter.
2. If needed, select an edge device group for this device from the *Group* list.
3. Select **AlienDevice** from the *Driver* list and then click **Go**. The parameters specific to the Alien Device driver appear.
   - Set *IPAddress* to the hostname or IP address of the machine running the Device Controller. If it runs on the same machine as the Oracle Sensor Edge Server, enter *127.0.0.1*.
   - Enter the port number used to communicate with the device (23 is the default).
   - Set the *Username*.
   - Set the *Password*.
   - Set the `AntennaSeqIDList` to the list of identifiers for each antenna.
   - Set `AntennaMappedDeviceNameList` to the list of mapped device names associated with each antenna.
4. Click **Finish** to complete the device.
5. Restart the Oracle Sensor Edge Server to receive events.

4.3.3.3 Configuring the IntermecDevice Driver

The IntermecDevice driver supports all of the RFID readers by Intermec, including the OEM Reader (Microwave, UHF), the PC Reader (PCMCIA), and the Fixed Reader (Serial or Ethernet). Other Intermec readers that support the Intellitag IDK also work with the IntermecDevice driver. For more information, refer to

http://www.intermec.com

Requirements

The IntermecDevice requires the following components, which are bundled and shipped with the Intermec driver:

- IntelliTag IDK

  The IntelliTag IDK (the IDK) is a set of Intermec-supported software libraries and tools. This library, which is the only supported method of communicating with

Intermec devices, is supported only on the Windows 32 platform (that is, Windows 2000 and Windows XP). The IntelligTag IDK is available at

http://www.oracle.com/technology/products/iaswe/edge_server/extensions.html

- Oracle Sensor Edge Server Device Controller

  The Oracle Sensor Edge Server Device Controller (the Device Controller) communicates with the local IDK API and exposes a network protocol that enables the Oracle Edge Server to communicate with the IDK.

- IntermecDevice Driver

  The IntermecDevice driver is the counterpart to the Device Controller, as it communicates with the Device Controller to drive the underlying devices.

The Oracle Sensor Edge Server can run on the same server as the Device Controller and the IDK, or on a separate server. Because the Intermec hardware exposes a Windows 32- based API, you must run the Oracle Sensor Edge Server on a Windows box or dedicate another Windows machine to only the Device Controller and the IDK.

Configuration Steps

Configuring the IntermecDevice driver involves the following tasks:

- Installing the IDK (Section 4.3.3.3.1).
- Registering Serial and PCMCIA readers (Section 4.3.3.3.2).
- Configuring the Intermec readers (Section 4.3.3.3.3).
- Using the Intermec tools to test the Intermec readers (Section 4.3.3.3.4).
- Installing the Device Controller (Section 4.3.3.3.5).
- Starting the Device Controller for the Intermec reader (Section 4.3.3.3.6).
- Creating a device using the IntermecDevice driver to enable communication between the Oracle Sensor Edge Server and the Device Controller for the Intermec reader (Section 4.3.3.3.7).

4.3.3.3.1 Installing the IntelliTag IDK To install the IDK and tools:

1. Connect the RFID reader to either a serial port, the PCMCIA slot of a PC, or a network segment. Connect a serial reader with a null modem (2/3 swap) cable. For more information, refer to the Intermec documentation.
2. Extract the content of the compressed file to a temporary directory, such as `(c:\temp):ORACLE_HOME\edge\lib\IDK_Beta_4.0.1.tgz.` (You can also download the latest version of the IDK from Intermec's Web site)
3. Install the IDK by running the install at `C:\temp\setup.exe`.
4. At the installer's first page, select **Next**.
5. In the Agreements page, read the license agreement and select **Yes** or **No**. Selecting **No** prevents you from proceeding.
6. On the next page, enter your name, the company name, and then select **Anyone who uses this computer**. Click **Next**.
7. Select a directory for the IDK. Use the default, if possible.
8. Select **Typical** for the setup type and then click **Next**.
9. Click **Next** to install the IDK and programs to the PC.

4.3.3.3.2 Registering the Serial and PCMCIA Readers You must register devices for the Serial and PCMCIA readers. The following steps to register these devices to do not apply to Ethernet readers.

To register devices:

1. From the *Start* menu, select **Intellitag IDK** and then **Device Registry Application**.
2. Select the *Register Readers* tab. Be sure that the reader is connected.
3. Select an existing reader from the *Select Reader* drop-down list or enter the name of a new reader in the *New Reader Name* field and then click **Register New Reader**.
4. In the *Port Name* field, enter the name of the port that you use to connect to the reader.

Accept the default settings (unless you have changed the device).

4.3.3.3.3 Configuring Readers Once you register a reader, you next configure it by editing the `rfconfig.ini` file. Open the `rfconfig.ini` file from the *Start* menu, select **IntelliTag IDK** and then **rfconfig.ini**. The file, which opens in Notepad is formatted as a standard Windows INI file. Each section of the file represents a new reader configuration, as illustrated by the `[Reader_One]` section in Example 4–4.

Example 4–4 rfconfig.ini

```
[Reader_One]
RFID_SWTT_FILE_NAME=C:\Program Files\Intermec\Intellitag IDK\swtt.ini
RFID_ATTR_TYPE=IT500 UAP Reader
IT500_PORT_TYPE=TCPIP
IT500_PORT_NUMBER=6543
IT500_CONNECT_TRIES=1
IT500_PORT_NAME=192.168.200.47
IT500_DEBUG_FILE_NAME="c:\IT500_Reader.log"
IT500_ANTENNA_TRIES=5
IT500_ANTENNAS=1 2 0 0 0 0 0 0
IT500_READ_TRIES=5
IT500_WRITE_TRIES=5
IT500_INTERR_DEBUG=0
IT500_READER_DEBUG=0
dll_name=C:\Program Files\Intermec\Intellitag IDK\it500.dll
IT500_IDENTIFY_TRIES=1
IT500_INITIALIZATION_TRIES=1
IT500_SIM_TAGS=5
IT500_IDENTIFY_READ_END_ADDR=17
IT500_HARDWARE_TYPE_CHECK=0
IT500_AUTOID_TIMEOUT=20
```

Although you can rename `[Reader One]`to any name, note this name for future reference. Modify (or verify) the following settings for the `rfconfig.ini` file:

- `IT500_PORT_TYPE`

  This parameter tells the API the type of connection to use, such as TCPIP for a network reader or ral for a serial or PCMCIA reader.

- `IT500_PORT_NAME`

  If it is a serial or PCMCIA reader, this parameter sets the name of the reader, that you registered (see Section 4.3.3.3.2). For network readers, this is the hostname or IP address of the reader.

- `IT500_PORT_NUMBER`

  This parameter specifies the TCP/IP port used to connect to the reader. The default setting is 6543. This parameter should only be defined for a network reader.

- `IT500_ANTENNAS`

  This is a mask for the antennae that are active and connected to the reader. The first digit corresponds to the first antenna. For example, if you have Antennas 1 and 3 connected to the reader and Antenna 1 is the first antenna, then set the parameter to `IT500_ANTENNAS=1 0 3 0 0 0 0 0`. For four antennae connected consecutively, set this parameter to `T500_ANTENNAS=1 2 3 4 0 0 0 0`.

  Save the Notepad file and then close it after you complete the configuration.

4.3.3.3.4 Testing the Readers To test the reader using Intermec tools:

1. From the *Programs* folder of the *Start* menu, click **Intellitag IDK** and then **RF Tag Map**.
2. Click **Select** to select a reader configuration. A dialog box appears listing the configurations defined in the `rfconfig.ini` file.
3. Select a reader configuration and then click **Select**. The **Open** button is activated.
4. Click the **Open** button to connect to the device
5. If then reader is connected properly, then the buttons in the *Tag Map* section are enabled.
6. Click **Start** to start the reader.
7. Wave the sample tags in front of the antenna. The tag ID and payload should be read and appear on the screen.

4.3.3.3.5 Installing the Oracle Sensor Edge Server Device Controller Install the Device Controller by extracting the

`ORACLE_HOME\edge\controller\deviceController.zip` file into the `C:\` directory. This extracts the Device Controller files into the `C:\controller` directory.

4.3.3.3.6 Starting the Oracle Sensor Edge Server Device Controller To start the device controller:

1. Start a command-line console (`cmd.exe`)
2. Navigate to the `C:\deviceController` directory and then run the following command:

   ```
   startIntermec.bat <ReaderName> <Port>
   ```

   where `<ReaderName>` is the is the configuration name of the reader in the rfconfig.ini file and `<Port>` is the port on which the IntermecDevice driver listens so that it can communicate with this Device Controller.

   For example, to start the Device Controller for the reader called Penn_A at port 6666, run the following command:

   ```
   startIntermec.bat Penn_A 6666
   ```

   After the Device Controller for the reader starts, create a device from the IntermecDevice driver that enables the Oracle Sensor Edge Server to communicate to the reader through this Device Controller.

4.3.3.3.7 Configuring the Oracle Sensor Edge Server to Communicate with the Device Controller You must create a device, an instance of the IntermecDriver driver, to enable the Oracle Sensor Edge Server to communicate with the Device Controller. To create a device:

1. Enter the name of the device. This is a mandatory parameter.
2. If needed, select an edge device group for this device from the *Group* drop-down list.
3. Select **IntermecDevice** from the *Driver* drop-down list and then click **Go**. The parameters specific to the IntermecDevice driver appear.
   - Set *IPAddress* to the hostname or IP address of the machine running the Device Controller. If it runs on the same machine as the Edge Server, enter *127.0.0.1*.
   - Set *PortNo* to the port number used to start the Device Controller (*6666* is the default).
   - Set the `AntennaSeqIDList` to the list of identifiers for each antenna.
   - Set `AntennaMappedDeviceNameList` to the list of mapped device names associated with each antenna.
4. Click **Finish** to complete the device.
5. Restart the Oracle Sensor Edge Server to receive events.

4.3.3.4 Configuring the Patlite Driver

Unlike the RFID readers or other sensors, the Patlite series of lightstacks and trees do not generate events, but instead act as indicator lights and signals. Sending events to Patlite lightstacks and trees turns on lights or causes them to blink for certain intervals.

Configuring the Patlite driver involves the following tasks:

- Installing the Patlite hardware (Section 4.3.3.4.1).
- Configuring the Device Controller for the Patlite device (Section 4.3.3.4.2).
- Starting the Device Controller for the Patlite device (Section 4.3.3.4.3).
- Configuring the Oracle Sensor Edge Server to communicate with the Device Controller for the Patlite device by creating a device instance (Section 4.3.3.4.4).

Supported Patlite Devices

Patlite’s products include those that support both Serial and Ethernet connection. The current version of the Patlite driver in this release supports the Serial connection only.

4.3.3.4.1 Installing the Patlite Hardware To connect the Patlite device, you must have the following hardware:

- A free RS232C communication port
- A Female/Female, nine-pin RS232 cable with a straight-through pin type (such as a modem cable).

To set up the hardware:

1. Connect the lightstack to a power supply.
2. Connect one end of the serial cable to the lightstack.
3. Connect the other end of the serial cable to the serial port.

4.3.3.4.2 Configuring the Oracle Edge Server Device Controller for the Patlite Device After you install the Device Controller (as described in Section 4.3.3.3.5), edit the `deviceController/config/dcconfig.xml` file as follows:

- Change the comName parameter to the com port that you are using.
- If the default port is currently in use on the local machine, change the value for lcPort to an available TCP/IP port number.

  In Example 4–5, the dcconfig.xml file uses COM3 as the value for comName and the default port of `7878` for the `lcPort` parameter.

Example 4–5 Editing the Com Port in dcconfig.xml

```
<?xml version="1.0"?>
<Configuration>
   <ConfigParam name="lcPort" value="7878" />
   <ConfigParam name="comName" value="COM3" />
</Configuration>
```

4.3.3.4.3 Starting the Device Controller for the Patlite Device To start the device controller:

1. Navigate to `deviceController/deploy/win`.
2. Run `startLight.bat`. A message similar Example 4–6 appears.

Example 4–6 Status Message for the Patlite Device Controller

```
C:\deviceController\deploy\win>startlight
Local ip is: 144.25.168.146
Establishing the listener at port: [7878] ...
Waiting for connections...
```

After the Device Controller starts, you can enable communication between the Oracle Sensor Edge Server and the Device Controller for the Patlite device by creating a device from the Patlite Device driver.

4.3.3.4.4 Configuring the Oracle Sensor Edge Server to Communicate with the Device Controller for the Patlite Device You must create a device, an instance of the Patlite Device driver, to enable the Oracle Sensor Edge Server to communicate with the Device Controller. To create a device:

1. Enter the name of the device. This is a mandatory parameter. For example, enter *Light1*.
2. If needed, select an edge device group for this device from the *Group* drop-down list.
3. Select *EdgeDevice* from the *Driver* drop-down list and then click **Go**. The parameters specific to the driver appear.
   - Select **Invoke Method**.
   - Set *IPAddress* to the hostname or IP address of the machine running the Device Controller.
   - Set *PortNo* to the TCP/IP port number set in the `dcconfig.ini` file (`7878` is the default).
4. Click **Finish** to complete the device.
5. Restart the Oracle Sensor Edge Server to run this device.

4.4 Managing Filters

A filter is a class that strains out unwanted events or translates higher level events from groups or events or specific conditions. An event is a message that is sent from either a sensor device or an application that signals that a state has changed. The Oracle Sensor Edge Server, which receives the data from these sensor devices or applications, normalizes the contents of these event messages by putting them in a common data format and then applies filters to strip them of extraneous information or unwanted events.

Filters can be attached to either a specific device or to a device group. Some filters are written as group-level filters and can only be attached to a device group. Likewise, some filters are written only for device-level filtering and only function if they are attached to a specific device. The filter object implements three levels of filtering:

Pre-Device Filtering

Pre-device filtering provides filtering against a batch of pass-in events before they are routed to the Oracle Sensor Edge Server device.

Post-Device Filtering

Post-device filtering provides any filtering against the events before they are merged up to a device group.

Device Group Filtering

Device group filtering provides filtering against events before they are delivered to an edge client.

> **Note:** Pre- and post-device filters apply only to devices; device group filtering applies only to device groups.

Applying Filters to Devices and Device Groups

A filter instance, which is an object of a filter, enables device groups and devices to use filters. Whenever a filter is applied to a device (or to a device group), a filter instance of that filter is created. For more information on device- and device-group filters, see Table 4–7.

The filter browsing screen enables you to create, edit and delete filters. The screen includes a list (described in Table 4–4) that organizes the filters available to the devices and device groups.

Table 4–4 Elements of the Filter Browsing Screen

Element	Description
Name	The name of the filter.
Class Name	The name of the filter implementation class.
Version	The version of the filter implementation.
Description	A description of the filter.
Created Date	The date the filter was created.

The Oracle Sensor Edge Server enables you add filters that provide pre-device and post-device filtering.

Note: Only the filters that you create enable pre- and post-device filtering. For more information on developing filters, refer to the *Oracle Application Server Wireless Developer's Guide*.

4.4.1 Adding a Filter

To create a filter, first click **Create**. The *Create New Filter* screen appears. Using the **Import** button, locate and then upload the filter's Extension Archive file to the repository. The Extension Archive file is a JAR file containing all of the class files and native binaries of the filter, as well as its properties files or static data files. In addition, the Extension Archive includes the Extension Archive Descriptor file, an XML file containing instructions for the Oracle Sensor Edge Server on loading and managing the filter.

Once the filter's Extension Archive file has been uploaded, the parameters specific to the uploaded file appear. (Unless the Extension Archive Descriptor file has been modified, the filter's parameters are read-only.) Click **Finish** to complete the filter. The filter, which then appears in the browsing screen, is available to devices and device groups. For more information on applying a filter to a device, see Section 4.6.

4.4.2 Deleting a Filter

To delete a filter from the repository, first select a filter from the list and then click **Delete**. Deleting a filter results in the deletion of the filter instances of that filter.

4.5 Managing Dispatchers

Dispatchers forward events from the Oracle Sensor Edge Server to either a dispatching layer or directly to an application.

The dispatcher browsing screen, which you access by selecting the *Dispatchers* tab, enables you to create and delete dispatchers. The following sections describe these functions:

- Section 4.5.1, "Setting a Dispatcher"
- Section 4.5.2, "Deleting a Dispatcher"

The Dispatchers browsing screen includes a list (described inTable 4–4) that organizes the dispatchers available to the Oracle Sensor Edge Server processes.

Table 4–5 Elements of the Dispatcher Browsing Screen

Element	Description
Name	The name of the dispatcher.
Class Name	The name of the dispatcher implementation class.
Version	The version of the dispatcher implementation.
Description	A description of the dispatcher.
Created Date	The date the dispatcher was created.

4.5.1 Setting a Dispatcher

To set a dispatcher, first click **Create**. The *Create New Dispatcher* screen appears. Using the **Import** button, locate and then upload the dispatcher's Extension Archive file to the repository. The Extension Archive file is a JAR file containing all of the class files and native binaries of the dispatcher, as well as its properties files or static data files. In addition, the Extension Archive includes the Extension Archive Descriptor file, an XML file containing instructions for the Oracle Sensor Edge Server on loading and managing the dispatcher.

Once the filter’s Extension Archive file has been uploaded, the parameters specific to the uploaded file appear. (Unless the Extension Archive Descriptor file has been modified, the filter’s parameters are read-only.) Click **Finish** to complete the dispatcher. The dispatcher, which then appears in the browsing screen, is among the dispatchers available when you create an edge dispatcher for an Oracle Sensor Edge Server process. For more information on creating an edge dispatcher, see Section 4.8.2.

> **Note:** The readme of Edge Developer's Kit, available from the Oracle Technology Network (`http://www.oracle.com/technology/`), explains how to format dispatchers so that the Oracle Sensor Edge Server can automatically populate parameters from the uploaded JAR file.

4.5.2 Deleting a Dispatcher

To delete a dispatcher from the repository, first select a dispatcher from the list and then click **Delete**. Deleting a dispatcher also results in the deletion of the edge dispatchers (that is, the instances of that dispatcher).

4.6 Managing the Devices of an Edge Server Process

An edge device is an end point of a sensor- based architecture, such as a Radio-Frequency Identification (RFID) reader, a dry contact, an laser diode, carousel, scale, robotic picker, or indication devices such as light sticks or message boards. Sensors are hardware or software end points that make observations of certain changes in state. While this is usually a physical change, such as a laser diode detecting that something has blocked the line of its beam, it can also a software observation of a change occurring within software, such as when a monitor **daemon** running on an edge controller exits. Sensors also observe defects in software.

The Manage Edge Devices button on the *Edge Server List* table enables you to add a device to a selected Edge Server process, delete a device from an Oracle Sensor Edge Server process, or edit the properties of a device. You can also start or stop a device. The following sections describe these tasks.

- Section 4.6.1, "Adding a Device to an Oracle Sensor Edge Server Process"
- Section 4.6.2, "Managing the Filter Instances for a Device or Device Group"
- Section 4.6.4, "Editing Filter Instances"
- Section 4.6.5, "Deleting Filter Instances from Devices and Device Groups"
- Section 4.6.6, "Starting and Stopping a Device"

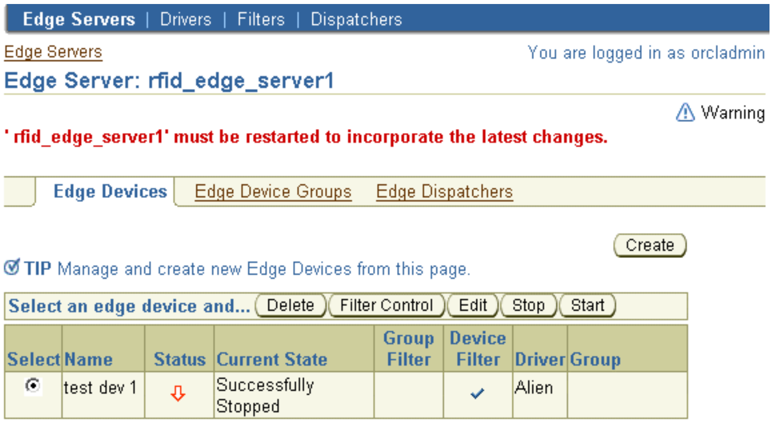
By selecting and an Oracle Sensor Edge Server process and then by clicking the **Manage Edge Devices** button, you access the *Edge Devices* tab, which includes a table listing of the devices assigned to the selected process (Figure 4–2). Table 4–6 describes the elements of this list.

Table 4–6 The Devices Table

Element	Description
Name	The name of the device.
Object ID	The Object ID of the device that is stored in the repository.
Status	Whether the device is running (indicated by an upward pointing arrow), or stopped (indicated by an downward pointing arrow).
Current State	The current state of the device.
Group Filter	A check mark indicates that this device belongs to a device group that is associated with a filter instance that strains out unwanted events
Device Filter	A check mark indicates that this device is associated with a filter instance.
Driver	The type of driver used by this device.
Group	The name of the device group to which this device belongs.

This table also includes buttons that enable you to delete a selected device from an Oracle Sensor Edge Server process, edit the properties of a device, manage the filter instances assigned to a device, as well as buttons to start and stop a selected device. The screen also includes the **Create** button, which enables you to add a device to the selected Oracle Sensor Edge Server process.

Figure 4–2 Managing the Devices



You are logged in as orcladmin

Edge Server: rfid_edge_server1

⚠ Warning

' rfid_edge_server1' must be restarted to incorporate the latest changes.

TIP Manage and create new Edge Devices from this page.

Select an edge device and...

Select	Name	Status	Current State	Group Filter	Device Filter	Driver	Group
●	test dev 1	⇩	Successfully Stopped		✓	Alien	

4.6.1 Adding a Device to an Oracle Sensor Edge Server Process

To add a device, first select an Oracle Sensor Edge Server process from the *Edge Server List* (Figure 4–1) and then click **Create**. The *Create New Device* screen appears, enabling you to add a device to the selected Oracle Sensor Edge Server process by defining the following values:

1. Enter the name of the device. This is a mandatory parameter.
2. If needed, select an edge device group for this device from the *Group* drop-down list.
3. From the *Driver* drop-down list, select the driver type for this edge device and then click **Go**. The parameters specific to selected edge driver appear.
4. Define the parameters of the driver as needed. For more information on drivers, see Section 4.3.1, "Adding a Driver".

5. Click **Finish** to complete the device. Click **Cancel** to clear all values and return to the device management screen.

4.6.2 Managing the Filter Instances for a Device or Device Group

A filter instance is an instantiated object of a filter. Whenever a filter is applied to a device (or to a device group), a filter instance is created, enabling the device or device group to use the filter. The **Filter Control** button, located in the tables listing the available devices or device groups associated with an Oracle Sensor Edge Server process, enables you to add, edit, delete, or arrange filter instances.

Although you can develop your own filters and then upload them (see Section 4.4.1), OracleAS Wireless includes a set of filters out of the box (described in Table 4–7).

Table 4–7 The Pre-Seeded Filters of the Oracle Sensor Edge Server

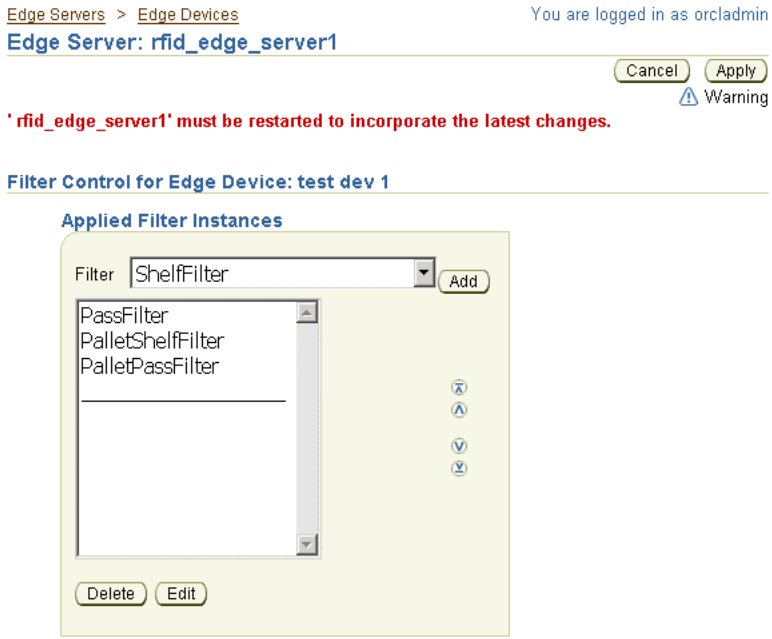
Filter Name	Function	Applied to Device Group? (Supports Group-Level Filtering)	Applied to Devices? (Supports Device-Level Filtering)
Check Tag ID Filter	A diagnostic tool that checks if a device is reading tags during a specified interval. See Section 4.6.3.1	No	Yes
Cross-Reader Redundant Filter	Blocks redundant events that are sent from the devices of a device group. See Section 4.6.3.2	Yes	No
Debug Filter	Tracks events passing through the system and then writes these events to a log file. See Section 4.6.3.3	No	Yes
Pass Filter	Notifies applications that a tag has passed through a device’s gateway or range or transmission. This filter also blocks events, so that only one event, rather than duplicate events, are generated when a tag is detected by a device. See Section 4.6.3.4	No	Yes
Shelf Filter	Signals that an item has entered, or exited the field or gateway of a device reader. See Section 4.6.3.5	No	Yes
Pallet Pass-Thru Filter	Enables you to see all of the tag IDs for items held in a container or on a pallet. See Section 4.6.3.6	No	Yes
Pallet Shelf Filter	Sends events that signal new containers or pallets entering or exiting the field or gateway of a device reader See Section 4.6.3.7	No	Yes

4.6.2.1 Adding a Filter Instance to a Device or to a Device Group

Adding a filter instance to a device enables pre-device filtering and post-device filtering of events. For a device group, a filter instance provides device group filtering. For more information on device- and device group-level filtering see Section 4.4.

To add a filter instance to a device, select a device from the browsing screen and then click **Filter Control**. The *Filter Control* screen appears, displaying the list for the selected device or device group in the *Applied Filter Instances* pane (Figure 4–3).

Figure 4–3 The Filter Control Screen



Edge Servers > Edge Devices

You are logged in as orcladmin

Edge Server: rfid_edge_server1

' rfid_edge_server1' must be restarted to incorporate the latest changes.

Filter Control for Edge Device: test dev 1

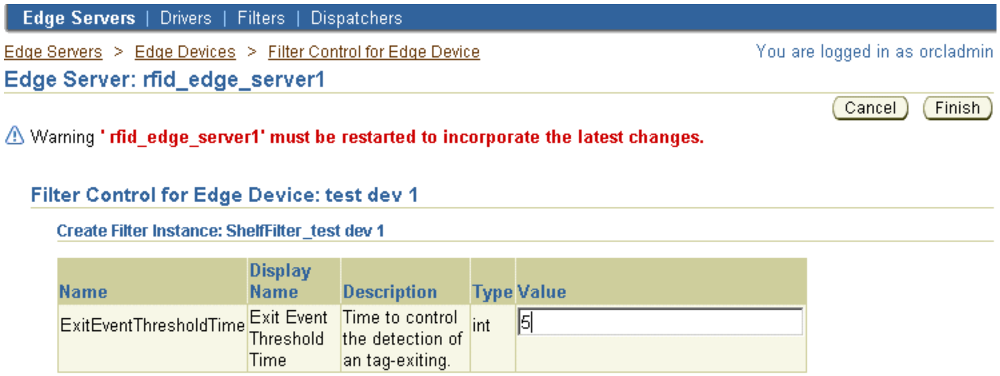
Applied Filter Instances

Filter: ShelfFilter

PassFilter
PalletShelfFilter
PalletPassFilter

Select a filter from the Filter drop-down list and then click **Add**. The parameters specific to the selected filter appear. Define the values as needed and then click **Finish** to create a instance of the filter for the selected device or device group. The filter instance appears in the *Applied Filter Instances* pane (Figure 4–3), where it can then be deleted, edited, or sorted.

Figure 4–4 Defining the Values of a Filter



Edge Servers > Edge Devices > Filter Control for Edge Device

You are logged in as orcladmin

Edge Server: rfid_edge_server1

Warning ' rfid_edge_server1' must be restarted to incorporate the latest changes.

Filter Control for Edge Device: test dev 1

Create Filter Instance: ShelfFilter_test dev 1

Name	Display Name	Description	Type	Value
ExitEventThresholdTime	Exit Event Threshold Time	Time to control the detection of an tag-exiting.	int	5

4.6.2.2 Prioritizing Filter Instances for Devices and Device Groups

The order of filter instances affects the efficacy of the data filtering. For example, if a device or device group is assigned a group filter, which groups IDs into an array of events and treats them as one item and a tag filter the filters out the events for a specific tag, *TagXYZ*, then applying the group filter before the tag filter results in the Oracle Sensor Edge Server receiving events grouped into chunks based on when they were detected, but only after the tag filter has strained out the events for *TagXYZ*. Reversing the order of the filters (that is, putting the group filter before the tag filter) would prevent the tag filter from filtering out anything, because it would see only the group events and not those for *TagXYZ*.

You can arrange the filter instances by selecting a filter instance and then moving it up or down using the arrow keys. To set the filter sequence, click **Finish**.

4.6.3 Defining the Parameters of the Pre-Seeded Filters

The following sections describe how the pre-seeded filters generate events and their configuration parameters:

- Section 4.6.3.1, "Configuring the Check Tag ID Filter"
- Section 4.6.3.2, "Using the Cross-Reader Redundant Filter"
- Section 4.6.3.3, "Using the Debug Filter"
- Section 4.6.3.4, "Configuring the Pass Filter"
- Section 4.6.3.5, "Configuring the Shelf Filter"
- Section 4.6.3.6, "Configuring the Pallet Pass Thru Filter"
- Section 4.6.3.7, "Configuring the Pallet Shelf Filter"

4.6.3.1 Configuring the Check Tag ID Filter

A Check Tag is any normal tag used to test if the device (in this case, a reader) is reading tags. Because the Check Tag itself should be physically located within the field of the reader, it should always be read; when other tags move through the field of the reader, the device also reads the Check Tag in conjunction with them.

The Check Tag ID Filter ensures that the device reads a Check Tag periodically. Using this filter enables you to check the status of a driver, its corresponding reader, and attached antennae. Because the Check Tag ID Filter is used solely for diagnostic purposes, it does not provide any events for dispatching to client devices. Instead, this filter generates an event if it does not detect that the device has read a Check Tag for a specified period of time.

Note: You can apply the Check Tag ID Filter only to devices.

Table 4–8 describes the parameters (and associated values) of the Check Tag ID filter.

Table 4–8 Parameters of the Check Tag ID Filter

Name	Value Type	Description
Check Tag Id	A `String` value.	The tag ID of the Check Tag, which is the ID that the filter searches for to see if the tag is being read.
Tag Check Time Window	An `int` value.	The period of time, in milliseconds, after which an event is generated if the filter has not seen the specified Check Tag.

To define the parameters for the Check Tag ID Filter, you must note the ID of the Check Tag itself (which must be placed within the field of the reading device). Enter this ID as the String value of Check Tag Id. If the filter does not detect that a device has read a Check Tag bearing the specified ID for the period defined in the *Tag Check Time Window* parameter, it generates an event. Table 4–9 describes the signature of the generated event. Refer to Section 4.1.1 for more information on the Event type.

Table 4–9 The Event Signature of the Check Tag ID Filter

Event Field	Value
`sourceName`	This field identifies the originator of the event. This is an optional field; its value is set by the client.
`correlationId`	The client sets the value for this field, which is used for message responses to a particular client (such as checking if a device functions). Any message sent back by the client has the same ID. This is an optional field.
`siteName`	The name of the site that generated this event.
`deviceName`	The name of the device that generated this event.
`time`	The time that the filter generated this event.
`type`	The message type, `Event.MESSAGE_EVENT` Note: Applications should subscribe to this message type for notifications of when devices fail.
`subtype`	`Event.ERROR_REPORT` (This is the subtype of the message type.)
`id`	The ID of the Check Tag (this is the value defined in the filter's *Check Tag Id* parameter).
`data`	An additional error message (if any).

4.6.3.2 Using the Cross-Reader Redundant Filter

The Cross-Reader Redundant Filter blocks redundant events that are sent from the devices of a device group and does not generate any events. This filter considers events redundant if it finds they have the same tag ID.

The Cross-Reader Redundant Filter is for group-level filtering only; it performs no functions if applied to a device. This filter has no parameters to configure.

4.6.3.3 Using the Debug Filter

The Debug Filter traces events passing through the system. Upon receiving events from its associated device, this filter writes events to a log file. This filter has a single parameter called *Event Output File*. To define this parameter, enter the full path of the log file to which you want the Debug Filter to write events. (The server must make this file writable.) The format of the Debug Filter's output is as follows:

```
"Devicename: <devicename> Type: <type> Subtype: <subtype>
EventTime: <time>TagIds:<tagid(,tagid)*>Data:<dat(,data)*>\n"
```

Each event is on a separate line; each line is separated by a newline character (LF or CRLF, depending on the operating system). The `<time>` value is a long as returned by the time(2) call.

This filter can only be attached to a device, not to a device group.

4.6.3.4 Configuring the Pass Filter

When a tag passes through the range of transmission, or through the gateway of a device reader, it generates a series of "tag is seen" events. The device reports these events periodically, starting when the tag enters the transmission field. The reporting stops when the tag moves out of the **reader field**.

Applications often do not require the series of events that a device reader generates; instead, these applications only need to know that a tag has passed through a device's gateway or range of transmission. The Pass Filter applies to such situations, as it reduces all of the "tag is seen" events into single events for each unique tag that passes through the field of a reader device.

The Pass Filter has one parameter, *Exit Event Threshold Time*. To define this parameter, enter the time (an int value), in milliseconds, since the device last read the tag before it is considered to have moved out of the device's transmission field. The parameter settings, which range from 50 milliseconds to under two seconds, dictate the frequency (that is, the reader cycle) in which the device reports these "tag is seen" events. If you set this frequency too high, such as to two seconds, then the device may miss the tag altogether.

When the device first reads a tag, the Pass Filter caches the tag's ID (tagid), notes the time that the tagid was read into the cache, and then immediately sends the pass-through event. The filter blocks subsequent reads for this cached tagid. Each time the filter receives a new read from the device, it updates the time that it read the tagid into the cache. If the sum of the caching time and the value set for *Exit Event Threshold Time* is less than the current time, then the Pass Filter clears the tagid from the cache. The next time the device reads this tag, the filter considers it a new event, caches its tagid and sends out a new pass-through event. Refer to Section 4.1.1 for more information on the Event type.

Table 4–10 describes the signature of the pass through event.

Table 4–10 Signature of the Pass-Through Event

Event Field	Value
`sourceName`	This field identifies the originator of the event. This is an optional field; its value is set by the client.
`correlationId`	The client sets the value for this field, which is used for message responses to a particular client (such as checking if a device functions). Any message sent back by the client has the same ID. This is an optional field.
`siteName`	The name of the site that generated this event.
`deviceName`	The name of the device that generated this event.
`time`	The time that the event was generated.
`type`	`Event.OBSERVATION`
`subtype`	`Event.PASS`
`id`	The ID of the tag.
`data`	The data payload of the tag.

4.6.3.5 Configuring the Shelf Filter

The Shelf Filter is a device-level filter that generates events when a tag is detected within the field of a reader or when the tag has left the field. Like the Pass Filter, the Shelf Filter has a single parameter, *Exit Event Threshold Time*. To define this parameter, enter the time (an int value), in milliseconds, since the device last read the tag before it is considered to have moved out of the device's transmission field. Unlike the Pass Filter, however, the Shelf Filter silently clears its cache once the interval defined for the *Exit Event Threshold Time* parameter elapses and does not generate an event.

4.6.3.5.1 Events Generated by the Shelf Filter The Shelf Filter generates two events:

- Events Generated by the Shelf Filter
- Events Generated by the Shelf Filter

IN FIELD Event

The Shelf Filter generates this event (described in Table 4–11) when the device first detects the tag. See Section 4.1.1 for more information on the Event type.

Table 4–11 Signature of the IN FIELD Event

Event Field	Value
`sourceName`	This field identifies the originator of the event. This is an optional field; its value is set by the client.
`correlationId`	The client sets the value for this field, which is used for message responses to a particular client (such as checking if a device functions). Any message sent back by the client has the same ID. This is an optional field.
`siteName`	The name of the site that generated this event.
`deviceName`	The name of the device that generated this event.
`time`	The time that the Shelf Filter generated this event.
`type`	`Event.OBSERVATION`
`subtype`	`Event.INFIELD`
`id`	The ID of the tag.
`data`	The data payload of the tag.

OUT FIELD Event

The Shelf Filter generates this event (described in Table 4–12) when the interval defined for the *Exit Event Threshold Time* parameter elapses. See Section 4.1.1 for more information on the Event type.

Table 4–12 Signature of the OUT FIELD Event

Event Field	Value
`sourceName`	This field identifies the originator of the event. This is an optional field; its value is set by the client.
`correlationId`	The client sets the value for this field, which is used for message responses to a particular client (such as checking if a device functions). Any message sent back by the client has the same ID. This is an optional field.
`siteName`	The name of the site that generated this event.
`deviceName`	The name of the device that generated this event.

Table 4–12 (Cont.) Signature of the OUT FIELD Event

Event Field	Value
`time`	The time that the Shelf Filter generated this event.
`type`	`Event.OBSERVATION`
`subtype`	`Event.OUTFIELD`
`id`	The ID of the tag.
`data`	The data payload of the tag.

When a device first detects the tag, the Shelf Filter caches the ID of the tag and then generates an `IN FIELD` event. At this point, the tag is read during every reader cycle. While the tag may not be read during some of these cycles, it is read during others. When the device does not read the tag consistently for a period longer than that designated for the *Event Exit Threshold Time* parameter, then the filter removes the tag's ID from the cache and generates an `OUT FIELD` event. The devices stops reading the tag once it passes from the field of the device.

4.6.3.6 Configuring the Pallet Pass Thru Filter

The Pallet Pass Thru Filter collects all of the events received during a specified period and sends them out as a single event. When a pallet or container passes through a gateway or through the field of transmission of a reader device, this filter generates a single event for all of these tags. This filter enables you to see what items a container or pallet may hold.

The Pallet Pass Thru Filter includes the following parameters:

- Configuring the Pallet Pass Thru Filter
- Configuring the Pallet Pass Thru Filter

Exit Event Threshold Time

To define this parameter, enter the time (an `int` value), in milliseconds, since the device last read a tag before it is considered to have moved out of the device's transmission field. The parameter settings, which range from 50 milliseconds to under two seconds, dictate the frequency (that is, the reader cycle) in which the device reports these "tag is seen" events. If you set this frequency too high, such as to two seconds, then the device may miss the tag altogether.

Event Collect Control Time

To define this parameter, enter the time (an `int` value), in milliseconds, for a device to complete a reading cycle of the tags included in a pallet or container before starting a new reading cycle. When this time elapses, the reading cycle concludes (that is, the device has read all of the new tags) and the Pallet Pass Thru Filter then generates an event with the following signature (described in Table 4–13). Refer to Section 4.1.1 for more information on the Event type.

Table 4–13 Signature of the Pallet Pass Thru Event

Event Field	Value
`sourceName`	This field identifies the originator of the event. This is an optional field; its value is set by the client.

Table 4–13 (Cont.) Signature of the Pallet Pass Thru Event

Event Field	Value
`correlationId`	The client sets the value for this field, which is used for message responses to a particular client (such as checking if a device functions). Any message sent back by the client has the same ID. This is an optional field.
`siteName`	The name of the site that generated this event.
`deviceName`	The name of the device that generated this event.
`time`	The time that the event was generated.
`type`	`Event.OBSERVATION`
`subtype`	`Event.MULTIPLE_PASS`
`id`	A comma-separated list of tag IDs.
`data`	A comma-separated list of datum.

4.6.3.7 Configuring the Pallet Shelf Filter

The Pallet Shelf Filter collects all of the events received during a set interval and then sends them as a single event. This filter enables you identify when new containers or pallets holding many items enters an area, or exits the field or gateway of a device reader.

The Pallet Shelf Filter has the following parameters:

- Configuring the Pallet Pass Thru Filter
- Configuring the Pallet Pass Thru Filter

Exit Event Threshold Time

To define this parameter, enter the time (an `int` value), in milliseconds, from the last time that the device read the tag before it is considered to have moved out of the device's transmission field. The Pallet Shelf Filter silently clears its cache once the interval defined for the *Exit Event Threshold Time* parameter elapses and does not generate an event.

Event Collect Control Time

To define this parameter, enter the time (an `int` value), in milliseconds, for a device to complete a reading cycle for the tags of a pallet or container before starting a new reading cycle. When this time elapses, the reading cycle concludes (that is, the device has read all of the new tags) and the Pallet Shelf Filter then generates an event.

4.6.3.7.1 Events Generated by the Pallet Shelf Filter The Pallet Shelf Filter generates two events:

- Events Generated by the Pallet Shelf Filter
- Events Generated by the Pallet Shelf Filter

MULTIPLE IN FIELD Event

The Pallet Shelf Filter generates the MULTIPLE IN FIELD event when the device first detects the tags. This event has the following signature (described in Table 4–14):

Table 4–14 Signature of the MULTIPLE IN FIELD Event

Event Field	Value
`sourceName`	This field identifies the originator of the event. This is an optional field; its value is set by the client.
`correlationId`	The client sets the value for this field, which is used for message responses to a particular client (such as checking if a device functions). Any message sent back by the client has the same ID. This is an optional field.
`siteName`	The name of the site that generated this event.
`deviceName`	The name of the device reading the pallet or container that generated this event.
`time`	The time that the Pallet Shelf Filter generated this event.
`type`	`Event.OBSERVATION`
`subtype`	`Event.MULTIPLE_INFIELD`
`id`	A comma-separated list of tag IDs.
`data`	A comma-separated list of datum.

MULTIPLE OUT FIELD Event

The Pallet Shelf Filter generates the MULTIPLE OUT FIELD event when the interval defined for the *Exit Event Threshold Time* parameter elapses. This event has the following signature (described in Table 4–15):

Table 4–15 Signature of the MULTIPLE OUT FIELD Event

Event Field	Value
`sourceName`	This field identifies the originator of the event. This is an optional field; its value is set by the client.
`correlationId`	The client sets the value for this field, which is used for message responses to a particular client (such as checking if a device functions). Any message sent back by the client has the same ID. This is an optional field.
`siteName`	The name of the site generating this event.
`deviceName`	The name of the device reading the pallet or container that generated this event.
`time`	The time that the Pallet Shelf Filter generated this event.
`type`	`Event.OBSERVATION`
`subtype`	`Event.MULTIPLE_OUTFIELD`
`id`	A comma-separated list of Tag IDs.

Table 4–15 (Cont.) Signature of the MULTIPLE OUT FIELD Event

Event Field	Value
`data`	A comma-separated list of datum.

4.6.4 Editing Filter Instances

To edit a filter instance, select the filter instance from the *Applied Filter Instance* pane and then click **Edit**. The parameters for the filter instance appear. Modify the values as needed and then click **Apply** to commit the changes.

4.6.5 Deleting Filter Instances from Devices and Device Groups

To delete a filter instance from a device or from a device group, select the filter instance from the *Filter Instances* pane and then click **Delete**.

4.6.6 Starting and Stopping a Device

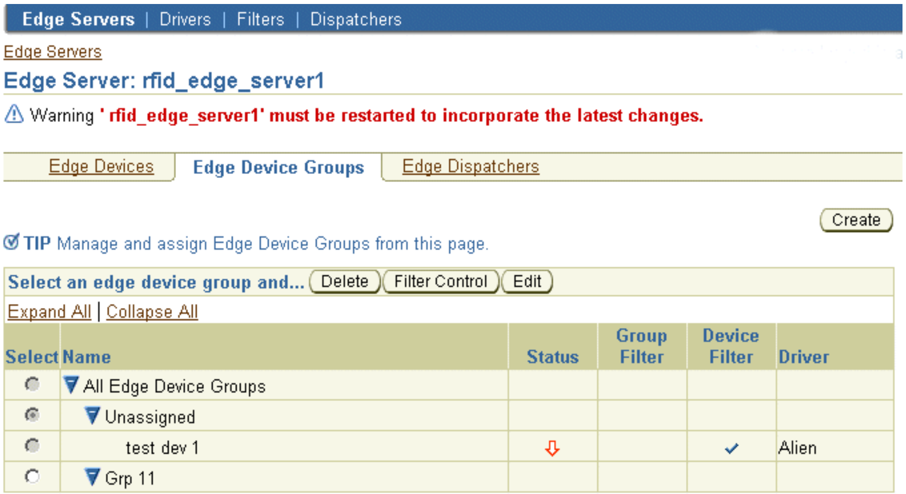
The **Start** and **Stop** buttons enable you to start or stop a device and its related filter instances. You do not need to stop a device before editing it.

4.7 Managing Device Groups

A device group is a logical grouping of devices. An Oracle Sensor Edge Server can have one or many device groups instantiated. Each device group is responsible for all of the devices (and their filters) included within it. Before you connect devices and filters to an Oracle Sensor Edge Server, you must first create device groups. The **Manage Device Groups** button on the *Edge Server List* table enables you to add a device group to a selected Oracle Sensor Edge Server process, delete a device group from an Oracle Sensor Edge Server process, or edit the properties of a device group (such as managing the device membership of the group). The following sections describe these tasks.

- Section 4.7.1, "Creating a Device Group"
- Section 4.7.2, "Editing a Device Group"
- Section 4.7.3, "Deleting a Device Group from an Oracle Sensor Edge Server Process"
- Section 4.7.4, "Managing the Filter Instances for a Device Group"

Figure 4–5 The Edge Device Groups Browsing Screen



Edge Server: rfid_edge_server1

⚠ Warning ' rfid_edge_server1' must be restarted to incorporate the latest changes.

TIP Manage and assign Edge Device Groups from this page.

Select an edge device group and...

Select	Name	Status	Group Filter	Device Filter	Driver
○	▼ All Edge Device Groups				
●	▼ Unassigned				
○	test dev 1	⇩		✓	Alien
○	▼ Grp 11				

By selecting and an Oracle Sensor Edge Server process and then clicking the **Manage Edge Device Groups** button, you access the *Edge Device Groups* browsing screen, which includes a table that lists the device groups assigned to the selected process (Figure 4–5). Table 4–16 describes the elements of this list.

Table 4–16 The Device Groups Table

Element	Description
Name	The name of the device group.
Object ID	The Object ID of the device group that is stored in the repository.
Status	Whether the devices belonging to the device group are running (indicated by an upward pointing arrow), or stopped (indicated by an downward pointing arrow).
Group Filter	A check mark indicates that this device group is associated with a group filter instance.
Device Filter	A check mark indicates that this devices belonging to this group are associated with filter instances.
Driver	The type of drivers used by the devices belonging to this group.

This table also includes buttons that enable you to delete a selected device group from an Oracle Sensor Edge Server process, edit the properties of a device group, and manage the filter instances assigned to a device group. The screen also includes the **Create** button, which enables you to add a device group to the selected Oracle Sensor Edge Server process.

4.7.1 Creating a Device Group

To create a device group, first select an Oracle Sensor Edge Server process and then click **Create**. The *Create New Edge Device Group* screen appears. Enter a name for the device group and then add devices to this device group by moving a device (or devices) from the *Available Devices* pane to the *Selected Devices* pane. Click **Finish** to complete the device group and add it to the selected Oracle Sensor Edge Server process.

You can group devices in terms of management if you want to treat them as a logical unit to manage, or you can group them by the type of filtering they perform. For example, if you group devices by cross-reader filtering, then you create a group of related devices and then attach filters to that group. For more information, see Section 4.6.2, "Managing the Filter Instances for a Device or Device Group".

Note: You must stop and restart the Oracle Sensor Edge Server process using the System Manager after you create, edit or delete a device group. For more information on starting and stopping a standalone instance, see Section 3.6.2.1, "Creating a Standalone Process".

4.7.2 Editing a Device Group

Selecting a device group and then clicking the **Edit** button enables you to change the name or the device membership of a device group. Clicking **Apply** commits any changes made to the device group. Clicking **Cancel** sets the device group back to its previous state.

4.7.3 Deleting a Device Group from an Oracle Sensor Edge Server Process

To delete a device group from an Oracle Sensor Edge Server process, select the device group and then click **Delete**.

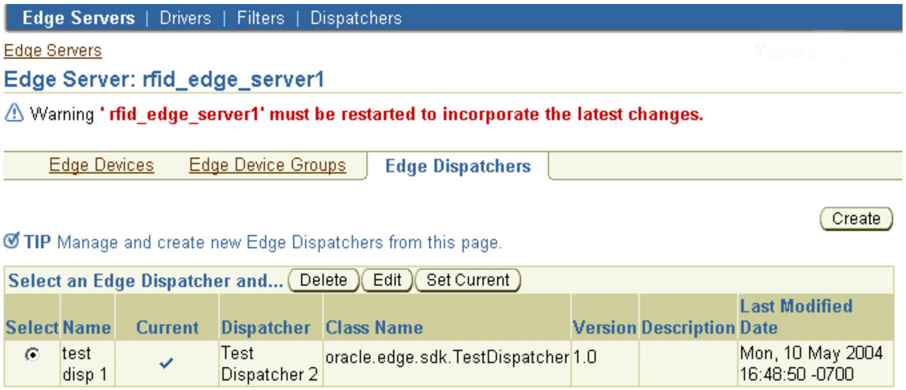
4.7.4 Managing the Filter Instances for a Device Group

See Section 4.6.2, "Managing the Filter Instances for a Device or Device Group".

4.8 Managing the Edge Dispatchers for an Oracle Sensor Edge Server Process

For an Oracle Sensor Edge Server process to use a dispatcher, you must assign an edge dispatcher to the process. An Oracle Sensor Edge Server process can use only one edge dispatcher. This edge dispatcher is noted as "Current".

Figure 4–6 The Edge Dispatcher Browsing Screen



Edge Server: rfid_edge_server1

⚠ Warning ' rfid_edge_server1' must be restarted to incorporate the latest changes.

TIP Manage and create new Edge Dispatchers from this page.

Select	Name	Current	Dispatcher	Class Name	Version	Description	Last Modified Date
●	test disp 1	✓	Test Dispatcher 2	oracle.edge.sdk.TestDispatcher	1.0		Mon, 10 May 2004 16:48:50 -0700

To access the functions to create and manage edge dispatchers for a selected Oracle Sensor Edge Server process, first select the Oracle Sensor Edge Server process and then click the **Manage Edge Dispatchers** button. The *Edge Dispatchers* browsing screen appears, which includes a table that lists the available edge dispatchers (Figure 4–6). Table 4–17 describes the elements of this list.

Table 4–17 Elements of the Edge Dispatcher Browsing Screen

Element	Description
Name	The name of the edge dispatcher.
Current	A check mark notes that the Oracle Sensor Edge Server process currently uses this edge dispatcher to forward events. An Oracle Sensor Edge Server process can use only one edge dispatcher.
Dispatcher	The name of the dispatcher used to create this edge dispatcher.
Class Name	The name of the dispatcher implementation class.
Version	The version of the dispatcher implementation.
Description	A description of the dispatcher.
Date Last Modified	The last time the edge dispatcher was updated.

The following sections describe the tasks enabled from the edge dispatcher browsing screen:

- Section 4.8.1, "Setting the Current Edge Dispatcher Used by the Oracle Sensor Edge Server Process"
- Section 4.8.2, "Setting an Edge Dispatcher for an Oracle Sensor Edge Server Process"
- Section 4.8.3, "Editing an Edge Dispatcher"
- Section 4.8.4, "Deleting an Edge Dispatcher from an Oracle Sensor Edge Server Process"

4.8.1 Setting the Current Edge Dispatcher Used by the Oracle Sensor Edge Server Process

An Oracle Sensor Edge Server process can use only one edge dispatcher at a time. To set the edge dispatcher for the Oracle Sensor Edge Server process, select an edge dispatcher and then click **Set Current**. Even if you have only one dispatcher configured for an Oracle Sensor Edge Server process, it can only be used by the dispatcher process if you set it as current; otherwise, the dispatcher will not be used. After you assign a dispatcher as current, the Oracle Sensor Edge Server process using it must be stopped and then restarted. For more information on starting and stopping a standalone process, see Section 3.6.2.1, "Creating a Standalone Process".

> **Note:** If you modify the assignments of devices, device groups, or dispatchers of an Oracle Sensor Edge Server process, or modify the properties of the devices, device groups or dispatchers used by a process, then you must stop and restart the Oracle Sensor Edge Server process to which these components belong. For more information on starting and stopping a standalone process, see Section 3.6.2.1, "Creating a Standalone Process"

4.8.2 Setting an Edge Dispatcher for an Oracle Sensor Edge Server Process

To set the edge dispatcher for an Oracle Sensor Edge Server process, click **Create**. The *Create New Edge Dispatcher* screen appears, in which you enter a name for the edge dispatcher, select the dispatcher used for this instance and then define the values for the parameters specific to the selected dispatcher. Clicking **Finish** completes the edge dispatcher.

You can select an edge dispatcher process to send event messages using Oracle Streams, Oracle's Java Message Service provider (OC4J JMS) remote Web Services or to a client application using HTTP.

4.8.2.1 Configuring the Edge Dispatcher to Use Oracle Streams

Configuring the edge dispatcher to use Oracle Streams and Advanced Queuing enables you to control how the edge dispatcher retrieves and distributes event messages. Unlike the OC4J JMS, Web Services, or HTTP dispatcher options, event messages dispatched using the Oracle Streams dispatcher do not have to be relayed directly to an entry point. The Oracle Streams dispatcher supports rule-based process and agent technologies. In addition, the Streams dispatcher only supports UTF-8 encoding.

Tips:

- Because Oracle Streams enables the propagation and management of data, transactions, and events in a data stream on one -- or across many -- databases, this dispatcher option provides the greatest flexibility of the seeded dispatcher options.
- The Oracle Streams dispatcher requires JDK 1.4.x.

Event messages are data that is deposited to a staging area (an internal queue). This data, in turn, can be aggregated in different ways for different client devices and applications (the consumers of the event messages). Using Oracle Streams as the dispatcher, the Data and Event layer of the database, not the Oracle Sensor Edge Server or applications, determines what an event is and when it is generated. The Data and Event layer provides a rule-based process that determines the appropriate event message for each client device or application.

Once the event messages are captured and placed into the staging queue, the event message data can be processed through the Rules Evaluation Job, which takes event messages from the staging queue and then compares them to the Oracle Sensor Edge Server rule set. Each rule has an action, which is executed if the rule applies. These actions include a PL/SQL callback for propagating the event message to other queues which could then be consumed by other applications. For more information on Oracle Sensor Edge Server Rule Set, refer to the *Oracle Application Server Wireless Developer's Guide*

In addition to these rule-based actions, the Rules Evaluation Job invokes applications by calling the Sensor Data Hub (SDH), which receives sensor data from the Oracle Sensor Edge Server or from other sources. The SDH includes the Sensor Data Archive, a set of archive tables that store all of the filtered sensor events in the system:

Note: Applications requiring raw, unfiltered event data that has not been processed by the rules can connect to the staging area using either AQ notification or JMS.

To configure the Streams dispatcher, select **Streams** and the define the following information connection information for the Streams staging area:

- The JDBC connection string to the database where the staging area resides. By default, this value is set to the IAS infrastructure database. Enter the URL to an application database providing optimal access and archiving of events and observations. The URL depends upon the driver type. For example, for a thin driver, enter `jdbc:oracle:thin@(description=(address=(host=<hostname>)<protocol=tcp)(port=<port>))(connect_data=(sid=<sid>)))`

where `<hostname>` is the host name or IP of the database server, `<port>` is the port number for the listener (1521, by default) and `<sid>` is the service id of the instance.

- The name of the user of the database where the staging area resides.
- The password to the user of the database where the staging area resides.

Select **Use Customer Database** to configure the Oracle Sensor Edge Server processes to send event messages through an external database.

Note: If you configure database streams as the event dispatcher method, you must perform the following post-installation steps:

1. Set up a database instance that can run Oracle Streams. This need not be the infrastructure database, but any database running applications. Be sure to note the connect string and password. You can use either the Standard or Enterprise Edition of the database as long as it is Version 9.2 or higher.
2. Enter the database password.
3. Log into any Wireless middle tier.
4. Go to the SQL directory (such as `ias_home/wireless/repository`).
5. Use SQL*Plus to connect to the database (Using the system account)
6. Run `create_edg_user.sql` to create a user for the Oracle Sensor Edge Server.
7. Enter the password you want to assign to the Oracle Sensor Edge Server schema.
8. Disconnect as system.
9. Reconnect as the Oracle Sensor Edge Server user using the database password.
10. Run `edg_create_streams.sql`.
11. Install the Sensor Data Hub (SDH) and then run (as the Oracle Sensor Edge Server user) `edg_create_sdh.sql`.

4.8.2.2 Configuring the Dispatcher to Send Messages Through OC4J JMS

OC4J JMS (OracleAS Containers for J2EE and Java Message Service), which is compatible with J2EE 1.3, is a Java Message Service based on Advanced Queuing (AQ) that provides guaranteed message delivery with auditing.

Note: In Oracle Application Server Wireless 10*g* Release 2 (10.1.2.02), the OC4J JMS dispatcher configuration (that is, the JMS dispatcher) cannot send messages back to sensor devices. Only the database stream dispatcher configuration supports this functionality.

To enable event message dispatching using OC4J JMS, you must configure a JMS topic queue called `edgeTopic` to which the dispatcher posts new event messages. In addition, you specify `edgeEventsConnectionFactory` as the connection factory. To enable the Oracle Sensor Edge Server components to access this topic, you must configure the jms.xml file under the OC4J container. Refer to the *Oracle Application Server Containers for J2EE Services Guide* for more information on configuring the JMS queue.

> **Note:** Upon startup, the JMS dispatcher looks for the `edgeTopic` queue using the JNDI (Java Naming Directory Interface) service implemented by the OC4J container. If the dispatcher cannot locate `edgeTopic`, it returns an error and then exits.

To create a JMS dispatcher that enables the Oracle Sensor Edge Server processes to send event messages to the JMS topic queue, `edgeTopic`:

1. Define the following values:
   - The URI of the OC4J instance where the edgeTopic queue is stored. This URI is used internally by OC4J ORMI to connect to the server. For example, enter `ormi://oc4j.us.oracle.com`.
   - The user name of the administrator of the OC4J instance where the `edgeTopic` queue is stored.
   - The password used by the administrator of the OC4J instance where the `edgeTopic` queue is stored.
   - Set the acknowledgement mode to `CLIENT_ACKNOWLEDGE` or `AUTO_ACKNOWLEDGE`. The default mode is `AUTO_ACKNOWLEDGE`.
   - Set `enterprise_mode` to *true*.
2. Perform the following steps for each middle tier that uses the JMS dispatcher:
   1. Add the `oc4j.jar` and `edgeclient.jar` libraries to the classpath section of *opmn.xml*, located in the `ORACLE_HOME/opmn/conf/`directory by adding the following lines under the process type with the ID of "edgeserver_server"::

      ```
      <environment>
      <variable id="CLASSPATH" value="$ORACLE_HOME/j2ee/home/oc4j.jar"
      append="true"/>
      <variable id="CLASSPATH" value="$ORACLE_HOME/wireless/lib/edgeclient.jar"
      append="true"/>
      <environment>
      ```

      > **Note:** The value for ORACLE_HOME depends on the location of an OracleAS Wireless middle tier.

   2. Copy `edgeclient.jar` to `ORACLE_HOME\wireless\lib\`. The `edgeclient.jar` is available from the Oracle Technology Network (http://www.oracle.com/technology/products/iaswe/edge_server/).
   3. Stop and then restart the OracleAS Wireless middle tier.

4.8.2.3 Configuring the Dispatcher to Send Event Messages to a Web Service

A client device or application can register a SOAP call which the Oracle Sensor Edge Server invokes when a new message must be delivered.

To configure the Web Service dispatcher to distribute event messages through Web Services, select **Web Services** and then define the parameters for this dispatcher type by entering the service URL of the WSDL (Web Service Definition Language) document that describes the client call. This URL must point to the EndPoint (port) of the Web Service. For example, enter `http://localhost:8888/wsdl/mytest.wsdl`. This document must contain the `portType` of `EdgeClientCallback` and the call, `processEvent`, as its child element. Upon startup, the Oracle Sensor Edge Server attempts to connect and bind to the service defined in this WSDL document.

4.8.2.4 Configuring the Dispatcher to Send Event Messages Through HTTP

Configuring the dispatcher to route events to clients using HTTP 1.0 results in the Oracle Sensor Edge Server posting each event message separately to the client. Because the Oracle Sensor Edge Server performs these posts sequentially, if one post is blocked, then all following posts are also blocked.

Select **HTTP** to configure the HTTP dispatcher and then enter the URL of the servlet, JSP, or CGI to which the Oracle Sensor Edge Server posts event messages whenever they are dispatched. To configure this dispatcher, enter the URL in the following format:

```
http://hostname:port/serverPath
```

If the Oracle Sensor Edge Server uses the HTTP dispatcher, then the client interface must tell the Oracle Sensor Edge Server how (and when) to call it.

Refer to the *Oracle Application Server Wireless Developer's Guide* for a description of the parameters posted by the Oracle Sensor Edge Server to the client application.

Click **OK** to complete the dispatcher configuration. Ensure that the selected dispatcher method functions after you complete this configuration. For more information on the configuring the dispatcher to route events through HTTP, refer to the tutorial, *Writing Sensor Based Applications Using JSP*, on the Oracle Technology Network (http:/www.oracle.com/technology/)

4.8.2.5 Using the Null Dispatcher

The Null dispatcher, which is created by default, discards all of the events passed to it. These events are not saved or spooled. Use this dispatcher only if you want to prevent the Oracle Sensor Edge Server from dispatching events.

4.8.3 Editing an Edge Dispatcher

By selecting an edge dispatcher and then clicking **Edit,** you access the *Edit Edge Dispatcher* page, which enables you to change the name of the edge dispatcher and modify the values of its parameters. You cannot select another dispatcher. Clicking **Apply** commits any changes made to the edge dispatcher. Clicking **Cancel s**ets the edge dispatcher's name and parameter values back to their previous state.

4.8.4 Deleting an Edge Dispatcher from an Oracle Sensor Edge Server Process

To delete an edge dispatcher from an Oracle Sensor Edge Server process, select the edge dispatcher and then click **Delete**.

5

Managing Users

This chapter includes the following sections:

5.1 Overview of User Management

The User Manager is a Web-based tool used to perform such user-support tasks as creating a new user, resetting the PIN and password for a user, assigning a special role to a user, or investigating any problems that a user may encounter when using the mobile applications. In the latter case, the User Manager enables you to view a user log, view and test user applications, and user devices.

The User Manager provides help desk functions for both developers and end users (OracleAS Wireless customers). In addition, the User Manager supports third-party content developers using Mobile Studio.

Note: Users granted the Super User or User Manager role access the User Manager tool. For more information, see Table 5–1.

5.1.1 Assigning User Roles

OracleAS Wireless users are assigned according to a user's responsibilities. These roles, which are described in Table 5–1, encompass all of the OracleAS Wireless resources, from server management and configuration, application development and publishing, help desk functions to subscription management.

Table 5–1 OracleAS Wireless User Roles

User Role	Description	Available Tools
Application Developer	Users assigned the Application Developer role perform the following functions: ■ Create, modify, delete and test applications. ■ Publish applications to the Application Developer's folder. ■ Create, modify, and delete notifications. ■ Create, modify, and delete data feeders. ■ Register and delete J2ME Web services. ■ Develop preset definitions.	Service Manager
Foundation Developer	Users assigned the Foundation Developer role perform the following functions: ■ Create, modify, and delete devices. ■ Create, modify, and delete transformers. ■ Create, modify, and delete regions. ■ Create, modify, and delete digital rights policies. ■ Create, modify, and delete API scan policies.	Foundation Manager
Content Manager	Users assigned the Content Manager role perform the following functions: ■ Manage application folders and bookmarks. ■ Create application links based on Application Developer-created applications. ■ Create notifications based on alerts (now deprecated). ■ Create application categories and associate access points with them. ■ Create a user-home folder rendering scheme, such as setting the sorting order for applications.	Content Manager
System Administrator	Users assigned the System role centrally manage and configure OracleAS Wireless.	The System Manager (accessed through the Oracle Enterprise Manager Application Server Control)

Table 5–1 (Cont.) OracleAS Wireless User Roles

User Role	Description	Available Tools
User Manager	Users assigned the User Manager role perform the following functions: ▪ Manage users by providing such Help Desk functions as editing a user profile, resetting passwords and PINs, and creating or deleting users. ▪ Manage user access privileges. ▪ View application links assigned to users. ▪ Manage user devices. ▪ Search for users. ▪ View overview information of users.	User Manager
Sensor Services Administrator	User assigned the Sensor Services Administrator role manage the devices and filters used with the Edge Server. These functions include: ▪ Create drivers for Oracle Sensor Edge Server devices ▪ Create filters used with Oracle Sensor Edge Server devices ▪ Manage the Oracle Sensor Edge Server devices assigned to the Oracle Sensor Edge Server processes	Sensor Services Tool
End User	Users assigned the end user role are the consumers of OracleAS Wireless applications. End-users create their own accounts when they register with OracleAS Wireless using the OracleAS Wireless Customization. End users can also customize their own services either from a desktop or from a device. Customization for end-users includes: ▪ Customize applications, download J2ME applications, subscribe to notifications. ▪ Manage devices. ▪ Manage location marks and location settings. ▪ Manage contact rules. Mobile studio users also have the end user role; a user belonging to the *StudioUser* group can access the Mobile Studio. Every OracleAS Wireless user is granted the Mobile Customer Role by default. This role is implicit to all users.	Wireless Customization Portal Mobile Studio (for users assigned to the *StudioUser* group)

OracleAS Wireless also allows anonymous users (the users who do not register with Wireless but want to use the applications as a guest). You can create an anonymous user account for each group. All unregistered users share the guest account to invoke applications owned by the group. A guest user cannot personalize applications.

5.1.2 Enabling Users to Access the Wireless Tools

You must assign roles to users from the User Manager rather than with other general-purpose user management tools, such as DAS (Oracle Delegated Administration Services). Users created using DAS or other OID (Oracle Internet Directory) tools are provisioned in OracleAS Wireless only when they accesses the Wireless and Voice portal, the mobile portal, or any of the PC-based tools for the first time. These provisioned users do not have the assigned roles needed to access the

OracleAS Wireless tools. For example, a user must have the Application Developer role to access the Service Manager. If a user with no assigned roles tries to log into a Wireless tool in the integrated mode, then OracleAS Wireless displays the following Single Sign-On (SSO) error:

No privilege to access this tool. Logout and login as another user with the required role.

The user can successfully log into the Wireless Tools (or other components) only after you assign that user a role. See Section 5.5 for information on creating a user and assigning user roles. For more information on creating users with OID, see the *Oracle Internet Directory Administrator's Guide*.

Note: SSO users of the Wireless Tools that have been deleted and then re-created cannot access some of the tools' functionality. The SSO server's caching of GUID information for users causes this problem, which can be solved by restarting the SSO server.

Note: There is a communication protocol between OracleAS Wireless and the OID server, synchronizing them when a user object is modified in one of the servers. In the previous version of OracleAS, version 1.1 of that protocol was used. This version supports protocols version 1.1 and 3.0. When you install OracleAS Wireless 10.1.2.0.1 by default you use version 1.1 of the protocol (this is to support mixed deployment environments in which some middle tiers are 10.1.2.0.0 and some are 10.1.2.0.1). A command line script ($ORACLE_HOME/wireless/bin/upgradeOIDProvProfile.bat/.sh) is providedso you can switch to the new protocol. There is one condition that should be met before upgrading to the new protocol: All middle tiers must be upgraded to 10.1.2.0.1; after running the script you can no longer add any more 10.1.2.0.0 middle tiers to your installation (but you can add more 10.1.2.0.1 middle tiers).

5.2 Logging In to the User Manager

Before using the User Manager, you must first access the login page for the OracleAS Wireless Tools using the following URL:

```
http://hostname:port/webtool/login.uix
```

For example, you access the login page by entering following URL into your browser:

```
http://hostname:7777/webtool/login.uix
```

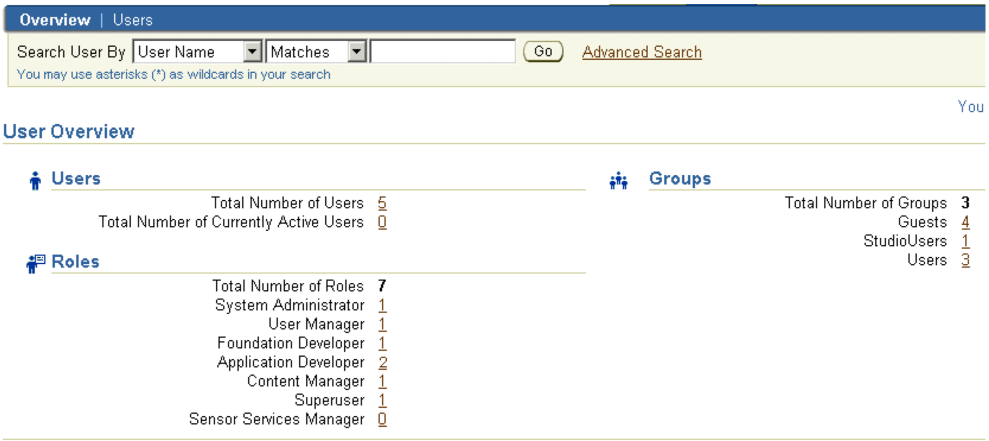
Note: 7777 is the default port number for Oracle Application Server Wireless. The port number range is 7777 to 7877. To ensure that you are using the correct port number, check the port number for Oracle Application Server Wireless stored in `[Oracle home]/install/portlist.ini`. For more information on port usage, see the *Oracle Application Server Administrator's Guide*

Enter your user name and then enter your password. If you are an administrator, enter *orcladmin* as your user name. (The password is set during installation, but can be changed with the User Manager.)

5.3 Using the User Manager

After you have successfully logged into the User Manager, the tool defaults to the User tab, displaying the User Overview screen.

Figure 5–1 The User Overview Screen



Overview | Users

Search User By User Name Matches Go Advanced Search

You may use asterisks (*) as wildcards in your search

You

User Overview

Users

Total Number of Users	5
Total Number of Currently Active Users	0

Roles

Total Number of Roles	7
System Administrator	1
User Manager	1
Foundation Developer	1
Application Developer	2
Content Manager	1
Superuser	1
Sensor Services Manager	0

Groups

Total Number of Groups	3
Guests	4
StudioUsers	1
Users	3

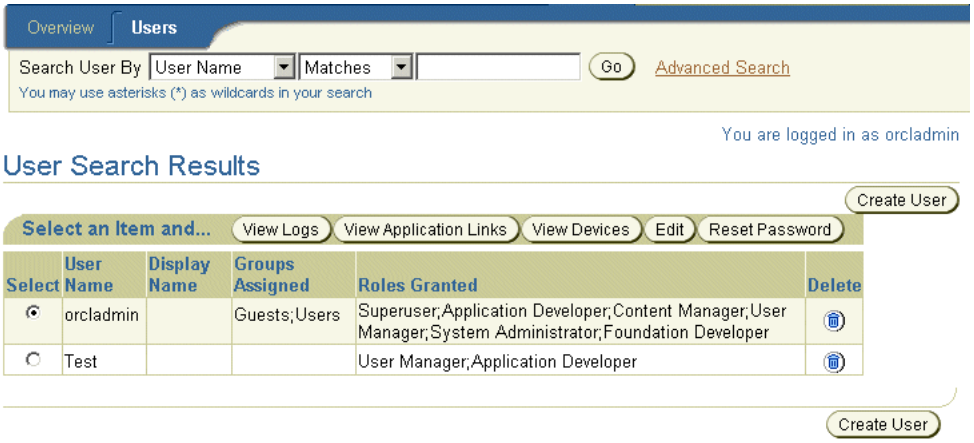
5.3.1 User Overview

The *Overview* tab provides you with an overall view of OracleAS Wireless users by displaying the following:

- Total number of OracleAS Wireless users.
- Total number of users with active OracleAS Wireless sessions.
- User Roles (A displays of the current user roles available and the number of users assigned to each of these roles.)
- User Groups (A display of the current OracleAS Wireless user groups and the number of users in each of these user groups.)

The User Manager displays the number of users as a link, which has a built-in query. Clicking this link invokes the *Users Search Results* page tab, which displays a table listing all of the users matching the selected link. For example, clicking *2* next to *Application Developers* in Figure 5–1 invokes the *Users Search Results* page illustrated in Figure 5–2, which displays the two OracleAS Wireless users who are assigned the Application Developer role.

Figure 5–2 The User Search Results Page



Search User By: User Name; Matches

You may use asterisks (*) as wildcards in your search

You are logged in as orcladmin

User Search Results

Select	User Name	Display Name	Groups Assigned	Roles Granted	Delete
●	orcladmin		Guests;Users	Superuser;Application Developer;Content Manager;User Manager;System Administrator;Foundation Developer	🗑
○	Test			User Manager;Application Developer	🗑

5.4 Searching for Users

Both the *Overview* and *Users* tabs contain a *Quick Search* area, in which you construct the criteria for finding current OracleAS Wireless users by specifying the user's name, display name, phone number, or e-mail address. You can further enhance your search by using the options accessed through the *Advanced Search* link, which enable you to find a user by user group, user role and user type.

Table 5–2 Elements of the Search User Result Screen

Label	Definition
User Name	The name of the user.
Display Name	The display name of the user.
Groups Assigned	The group to which the user has been assigned.
Roles Granted	The role (or roles) granted to the user. For information on user roles, see Section 5.5.

5.4.1 Finding Users with Quick Search

To find users using *Quick Search*, you first select the type of search by selecting from the following options in the drop-down list:

- User Name
- Display Name
- Phone Number
- Email Address

Enter the search string in the *Search By* field. For example, enter the user's display name in this field. From the drop-down list, select the match type (*matches*, *starts with*, *ends with*, or *contains*) and then click the **Go** button. The list of users corresponding to the search criteria appears.

If you want to specify more search options, such as finding a user belonging to a specific group, or with a specific role, click *Advanced Search* to further define your search.

5.4.1.1 Using Advanced Search

To find users using *Advanced Search*, you first select the type of search:

- User Name
- Display Name
- Phone Number
- Email Address.

Enter the search string in the *Search By* field. For example, enter the user display name in this field. Use the drop-down lists to select the group to which the user belongs, the role granted to the user, and the user type. There are three types of OracleAS Wireless users: anonymous users who are guest users, virtual users, and registered users. When an unregistered user accesses a OracleAS Wireless site, OracleAS Wireless detects the user and creates a virtual user account for that user.

> **Note:** The default value of the user role, user group, and user type search options is *Any*.

Click the *Search* button. The list of users corresponding to the search criteria appears in the Users subtab.

5.5 Creating Users

The *Create User* screen contains a set of parameters that administrators set to create and configure new users. Table 5–3 describes these parameters.

Table 5–3 Parameters of the Create User Screen

Parameter	Value
User Name	The name of the user. This is a required field. **Note**: the name is case-sensitive.
Display Name	The display name of the user.
New Password	The user's password. **Note**: the password is case-sensitive. This field is required when creating new user.
Password Confirmation	The user's password entered again. This field is required when creating new user.
Primary Phone Number	The primary phone number of the user. This is a required field. When you enter the value for this parameter, OracleAS Wireless creates a device for this user with this phone number. This number also serves as the user’s account number, which the user enters for login rather than entering a user name on a device.When you edit a user profile, the Account Number field in the editing screen corresponds to the *Primary Phone Number* field in the Create User screen. Changing the value for the *Account Number*, however, only changes the account number, not the value for the primary phone number.
PIN	The user's personal identification number (PIN) requested when the user logs in using the primary phone number (that is, the. account number). This field is required when creating a new user.
PIN Confirmation	The confirmation for the user's personal identification number. This field is required when creating new user.
Email Address	The user's e-mail address. This is a required field and it appears only in the Create User screen. A user device with this e-mail address is created for the user.
Mobile Station ID	The user's mobile phone number, or the MSISDN (mobile subscriber ISDN) for GSM (Global System for Mobile communication) services. OracleAS Wireless uses this ID to track the position of the user.

Table 5–3 (Cont.) Parameters of the Create User Screen

Parameter	Value
External Repository ID	A mapping of a user from the OracleAS Wireless schema to a unique ID of that user in another user database.
Virtual User Device ID	An ID generated for unregistered users who access a OracleAS Wireless site. When an unregistered user accesses a OracleAS Wireless site, OracleAS Wireless detects the user and creates a virtual user account for that user. Wireless traces this user by phone number or by another identification number sent from the user's device. This number is the Virtual Device ID number. You cannot create a virtual user; OracleAS Wireless creates virtual users dynamically. This parameter does not apply to registered users or to anonymous users.
User Type	Select *Registered* for registered OracleAS Wireless users. Select *Anonymous* when creating an anonymous user, an entity that OracleAS Wireless automatically assigns to an unknown user. An unknown user is a user whose device does not send any identifiable numbers through the HTTP header when accessing a OracleAS Wireless site. Creating an anonymous user enables unknown users to access public applications and explore the site before registering.
Gender	The user's gender (select either male or female).
Date of Birth	The user's date of birth. You can select this from the calendar or enter it in the field using the mm/dd/yyyy format.
Enabled	Selecting this check box enables users to log in. Leaving this check box clear prevents a user from logging in. By default, this option is enabled.
Language	A list of display languages. This is a required field.
Country of Residence	A list of countries used to set a user's selected country as the home or local area for calls. Because the device initiating the phone call (usually a VoiceXML gateway) may not be co-located with the OracleAS Wireless instance or even the user, having the user select a country of residence enables applications with such features as voice-activated dialing to treat the user-selected country of residence as the home or national area for calls. These applications, in turn, consider other areas as international calls. For the user, calls within the selected country of residence are local, while those outside of the selected country are not (regardless of whether the user's country of residence is the same as that of the device initiating the telephone call). For application developers, the country of residence can be correlated to a unique and uniformly accepted country code. All references to telephone numbers that must be passed to calling entities (that is, devices initiating the telephone call), can be formatted according to RFC 2806 (URLs for telephone calls), which stipulates a standard format for expressing all telephone numbers in an unambiguous fashion.
Time Zone	A drop-down of time zones for the user's locale. **Note**: OracleAS Wireless generates and delivers notifications to the time zone selected by the user rather than by the time zone of the OracleAS Wireless server itself. This is a required field.
User Home Root	A drop-down list of root folders, which can represent user communities or providers. The Content Manager creates these folders, which provision the home folders for users. This is a required field.
Remember my last location	This check box enables OracleAS Wireless to ascertain the user's current location using the signal from the user's mobile device, and cache the user's current location. Wireless sends the user content specific to the current location. Caching the location can improve server performance.
Allow other applications to access my identification.	This check box enables the user identity to be disclosed to a third-party application.

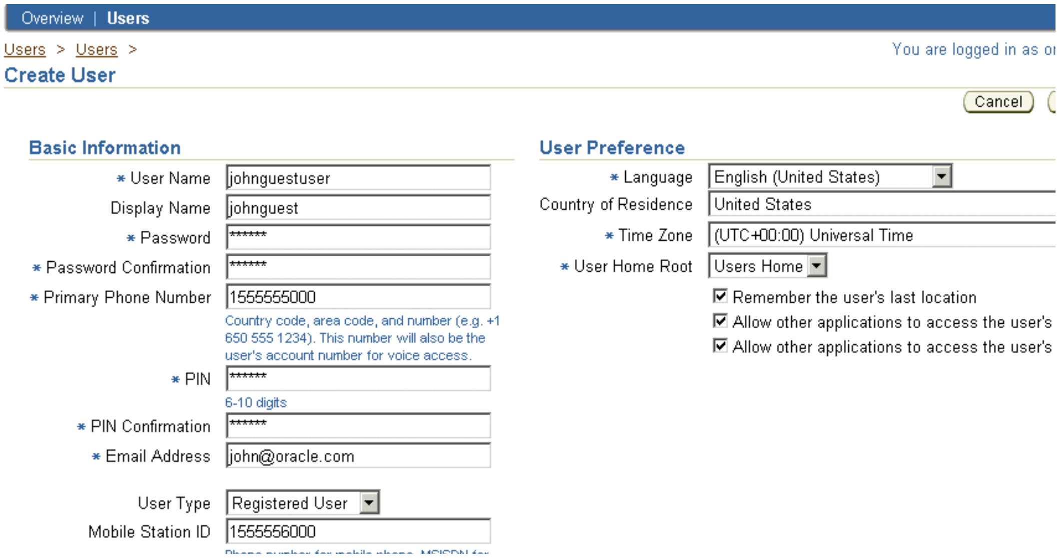
Table 5–3 (Cont.) Parameters of the Create User Screen

Parameter	Value
Allow other applications to access my location.	This check box enables the user's location be reported to a third-party application.
Groups	The groups to which you can assign the user. Using the arrow keys, you can select (> or >>) or remove (< or <<) a user from a group. The Content Manager creates groups and assigns applications to them. Refer to Section 6.5 for more information.
Roles	The roles to which you can assign a user. Using the arrow keys, you can select (> or >>) or remove (< or <<) a user from a role. If you do not select a role, then the user has end-user privileges and cannot access any OracleAS Wireless tool. A User Manager user can only create other User Manager users or end users.

Note: Users assigned the User Manager role only (that is, a user without the Super User privileges) can only assign the User Manager role.

To create a new user, you first click *Create User* in the Users screen. The create screen appears (Figure 5–3), with its fields populated by such default information as the user’s status as enabled and the default language and time zone (which are based on the corresponding configuration for the OracleAS Wireless Site). Enter the values as needed. The user name, password, primary phone number, PIN, e-mail address are required, as is the user’s language, time zone and User Home Root.

Figure 5–3 The Create User Screen (Partial View)



Create User

Basic Information

* User Name	johnguestuser
Display Name	johnguest
* Password	******
* Password Confirmation	******
* Primary Phone Number	1555555000 Country code, area code, and number (e.g. +1 650 555 1234). This number will also be the user's account number for voice access.
* PIN	****** 6-10 digits
* PIN Confirmation	******
* Email Address	john@oracle.com
User Type	Registered User
Mobile Station ID	1555556000 Phone number for mobile phone, MSISDN for

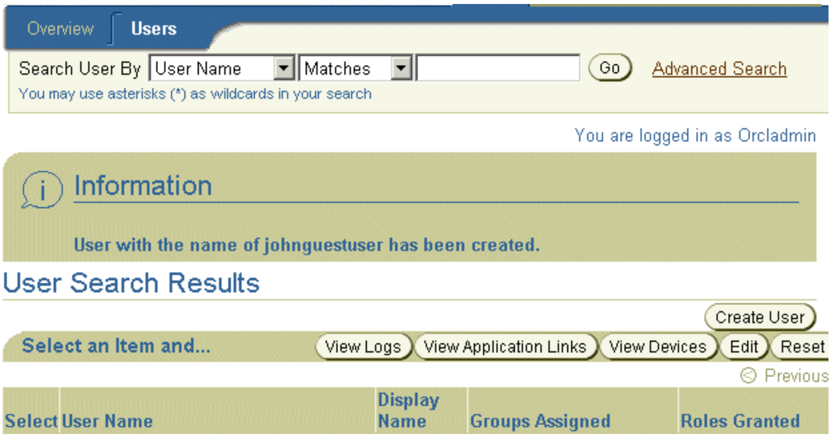
User Preference

* Language	English (United States)
Country of Residence	United States
* Time Zone	(UTC+00:00) Universal Time
* User Home Root	Users Home

☑ Remember the user's last location
☑ Allow other applications to access the user's
☑ Allow other applications to access the user's

Click *Finish* to complete the creation of the user. The new user appears in the user list in the Browse User screen, along with the message *User with the name of *** has been created*. Figure 5–4 illustrates the *Browse User* screen displaying this message.

Figure 5–4 The New User Message



Search User By User Name Matches

You may use asterisks (*) as wildcards in your search

You are logged in as Orcladmin

Information

User with the name of johnguestuser has been created.

User Search Results

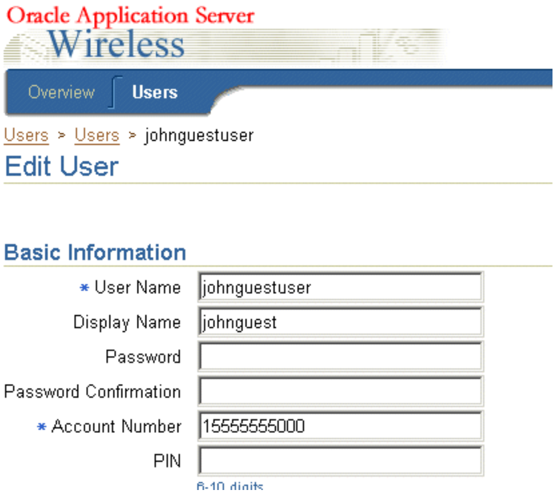
Select	User Name	Display Name	Groups Assigned	Roles Granted

Click *Finish* to complete the user. The new user appears in the user list in the Browse User screen, along with the message, *User with the name of *** has been created*.

5.5.1 Editing User Profiles

From the Users screen, select the user from the Users screen and then click **Edit**. The *Edit* screen appears, displaying the current user profile information for the selected user. When you edit a user profile, the *Account Number* field in the editing screen corresponds to the *Primary Phone Number* field in the *Create User* screen. For example, Figure 5–5 depicts a partial view of the *Editing* screen in which the value for the Account Number field, *1555555000* is the same as the value entered for the user's *Primary Phone Number* in Figure 5–4. Changing the value for the *Account Number*, however, only changes the account number itself, not the value for the primary phone number. You cannot edit the e-mail address for a user. To edit the values for the *Primary Phone Number* and for the e-mail address, you must edit the user's devices.

Figure 5–5 Editing a User Profile



Users > Users > johnguestuser

Edit User

Basic Information

* User Name	johnguestuser
Display Name	johnguest
Password	
Password Confirmation	
* Account Number	15555555000
PIN	
	6-10 digits

Edit the values as needed. See Section 5.5 for information on the parameters of a user's profile. The password and PIN are not required when editing user profiles, but you can edit these values if needed.

Click **Finish**. The browsing screen appears, displaying any changes pertinent to the labels in the *Users* screen (for example, the user name).

Note: A user assigned to the User Manager role (but not assigned to the Super User role) can only edit his or her own user profile, the user profiles for end users, and the profiles of other users assigned only to the User Manager role.

Users can view the Wireless Tools in 11 languages and the Wireless Customization in 29 languages. (The languages available for Wireless Customization include the 11 languages available to the Wireless Tools in addition to 18 more.)

5.5.2 Resetting the Password

The User Manager enables you to reset the user password and PIN.

From the *Browse User* screen, select a user and then click the **Reset Password** button. The Reset Password screen then appears, where you to enter the new password, the password confirmation, the new PIN, and the PIN confirmation. To reset the PIN only, do not enter a value *Password* field (leave it blank). Likewise, you can leave the *PIN* field blank if you need only to reset the password.

In the default installation of Oracle Application Server, a OracleAS Wireless application entity does not have the User Administrator privilege to change a user password, so saving the changed password fails with a general error message. You can identify the error by checking the OracleAS Wireless log file.

You can assign the User Administrator (`UserSecurityAdmins)` privilege by running the `assignUserSecurityAdminsPrivilege.sh` script, located at `ORACLE_HOME/wireless/bin`. To assign this privilege, run the command

```
assignUserSecurityAdminsPrivilege.sh cn=orcladmin welcome1
```

Note: `orcladmin` is the user name for the super user and *welcome1* is the password.

After running the script, you must restart the OID and OC4J_Wireless processes for the change to take effect.

For more information, refer to the *Oracle Application Server 10g Security Guide*. This privilege-checking applies only to password, not to PIN.

5.5.3 Deleting a User

To delete a user, select a user from the *Browse User* screen and then click the delete user icon. After confirmation, OracleAS Wireless deletes the user from the list.

5.6 Viewing Application Links

The **View Application Links** button enables you to view the applications, bookmarks, folders, and notifications accessible by a selected user, as well as to use the simulator to test applications. The applications that a user can access include all those assigned to the groups that the user belongs to, as well as applications created in the selected user's home folder using Mobile Studio, or published through the Service Manager using the **Quick Publish** function Table 5–4 describes the information displayed when you select a user and then click **View Application Links**.

Table 5–4 Application Link Information

Element	Description
Type	The type of objects created by the selected user.
Name	The display name of the folder, application, or bookmark.
Object ID	The Object ID (OID) of the application or module in the database.
Application	The master application on which the invoked the user's applications (that is, the application links published to the user’s group) are based.
Test	Clicking the phone icon enables you to view the application on a phone simulator.
Visible	If the column displays *true*, then the object is visible and therefore accessible to an end user. If *false*, then the object is not visible.
Sequence	The customized order in which applications and folders appear on output devices. By default, the display order of the applications is by name.
Group	The group to which the application is assigned.
Last Modified	The last time an object was modified.

For detailed information on the **Edit**, **Delete**, **Move** and **Debug** buttons, refer to Section 6.3. These functions are identical to those in the Content Manager, except that the User Manager only enables you to modify the attributes of applications which belong to the selected user's home folder; you cannot modify the applications assigned to the selected user’s group.

5.7 Viewing Devices

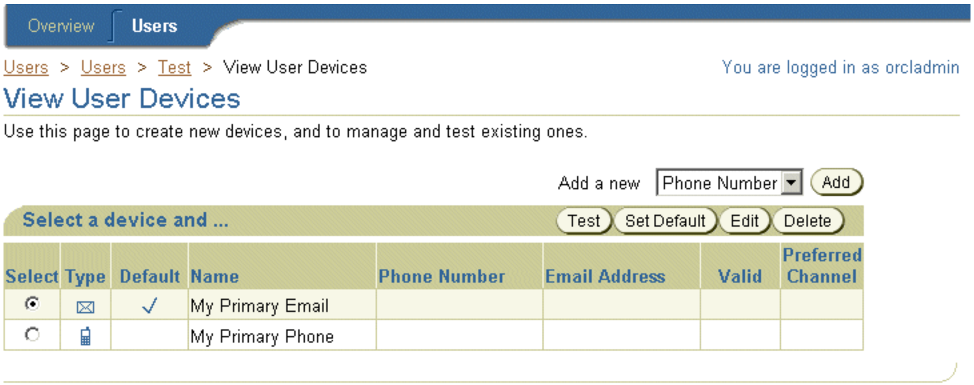
The User Manager enables you to manage a user's devices by clicking the **View Devices** button from the *Browse User* screen.

Clicking **View Devices** enables you to see all of the devices belonging to a selected user (as illustrated in Figure 5–6). The User Manager provides the same functionality as the OracleAS Wireless Customization Portal by enabling you to add, edit, delete, or validate a device, and set a default device.

The **Test** button enables you to test a selected device by sending a test message to the user. If a user cannot receive subscribed notifications, then this function indicates that there are problems with sending messages.

For more information on using the OracleAS Wireless Customization Portal for general device management and display attributes, refer to the *Oracle Application Server Wireless Developer’s Guide*.

Figure 5–6 Viewing User Devices



Users > Users > Test > View User Devices

You are logged in as orcladmin

View User Devices

Use this page to create new devices, and to manage and test existing ones.

Add a new Phone Number

Select a device and ...

Select	Type	Default	Name	Phone Number	Email Address	Valid	Preferred Channel
●	✉	✓	My Primary Email				
○	☎		My Primary Phone				

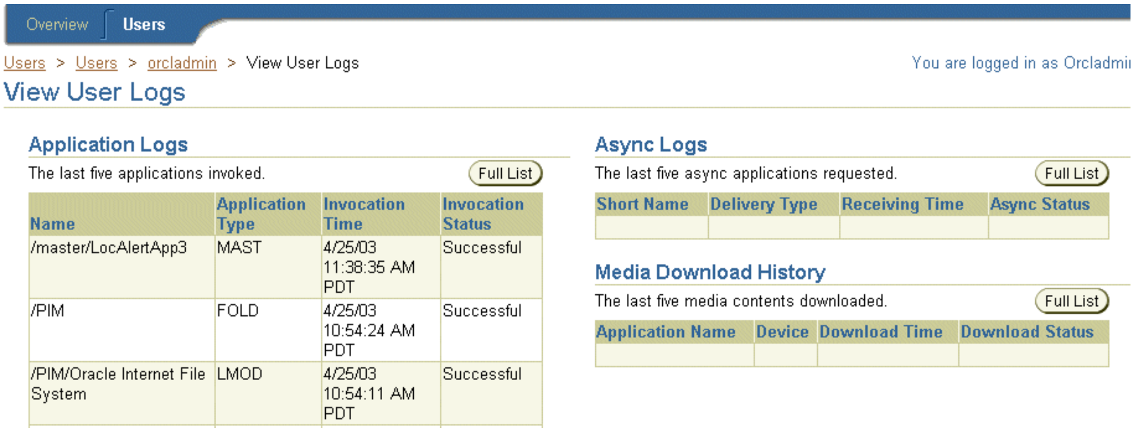
5.8 Viewing Logs

The User Manager enables you to view activity logs that display the accessed Async applications, notifications, applications, and the downloaded media contents (that is, J2ME applications) for a selected user. The activity logs display the most recent activity for user, or the user's activity within a specific time frame. In addition, these activity logs tell you if OracleAS Wireless dispatched applications successfully.

To view the user logs, select a user and then click the **View Logs** button. The summary page of activity log appears (Figure 5–7), displaying the last five logged records of the Async applications requested, notifications sent, applications accessed, and media contents downloaded.

To view the detailed activity based on a specified time frame, click the **Full List** button of the specific log type.

Figure 5–7 The Viewing User Logs Screen (Partial View)



Users > Users > orcladmin > View User Logs

You are logged in as Orcladmi

View User Logs

Application Logs

The last five applications invoked.

Name	Application Type	Invocation Time	Invocation Status
/master/LocAlertApp3	MAST	4/25/03 11:38:35 AM PDT	Successful
/PIM	FOLD	4/25/03 10:54:24 AM PDT	Successful
/PIM/Oracle Internet File System	LMOD	4/25/03 10:54:11 AM PDT	Successful

Async Logs

The last five async applications requested.

Short Name	Delivery Type	Receiving Time	Async Status

Media Download History

The last five media contents downloaded.

Application Name	Device	Download Time	Download Status

Viewing Async Logs

Table 5–5 describes the Async application statistics for a selected user.

Table 5–5 Async Log Statistics

Element	Description
Short Name	The name of the Async application (for example, *ST* for a stock quote application).
ID	The OID of the Async agent application in the database.
Device Address	The address of the user's device receiving the notification.
Server Address	The address of the Async application.
Delivery Type	The delivery type for the Async application (for example, SMS).
Receiving Time	The time that the Async server received the request.
Async Status	A message describing how Wireless failed to respond to the Async application.

Viewing Notification Logs

Table 5–6 describes the notification statistics for a selected user.

Table 5–6 Notification Log Statistics

Element	Description
Name	The name of the notification.
Notification ID	The OID of the notification in the database.
Device Address	The address of the user device receiving the notification.
Device Type	The type of device that receives the notification (for example, WAP-Push, SMS, or e-mail).
Dispatch Time	The time that OracleAS Wireless sent the message.
Message Status	The status of the sent message.

Viewing Application Statistics Logs

Table 5–7 describes the application statistics for a selected user.

Table 5–7 Application Log Statistics

Element	Description
Name	The name of the application.
Application ID	The OID of the application in the database.
Application Type	The type of object (folder, bookmark, application, or local module) accessed by the user.
Invocation Time	The time the user accessed the application.
Invocation Status	Whether Wireless successfully executed the application.

Viewing Media Download History Logs

Table 5–8 describes the media download statistics for a selected user.

Table 5–8 Media Download History Statistics

Element	Description
Application Name	The name of the J2ME application.
Content Version	The version of the deliverable content which was downloaded.

Table 5–8 (Cont.) Media Download History Statistics

Element	Description
Device	The name of the user device which downloaded the application.
Download Time	The time of the download.
Download Status	The status of the download.

5.8.1 Selecting a Time Frame

You can view the activity log for a specific period using the *From Date* and *To Date* fields. You can set the starting and ending dates either by entering them in the fields in the *mm/dd/yyyy* format, or by picking them from the calendars. Click **Go** after you have completed entering the date range.

> **Note:** The default *From* date is midnight of the previous day. Both the *From* and *To* dates assume midnight of the selected day

5.8.1.1 Printing an Activity Log

You can print an activity log by clicking **Printable Page**. This printed page contains text only and has no headers or footers. Use the browser's **Back** button to navigate from the printed page.

6 Managing Content

This chapter, through the following sections, describes the Content Manager.

- Section 6.1, "Overview of the Content Management"
- Section 6.2, "Accessing the Content Manager"
- Section 6.3, "Managing Application Links"
- Section 6.4, "Managing Notifications"
- Section 6.5, "Defining Access Control"
- Section 6.6, "Creating User Home Root Folders"
- Section 6.7, "Categorizing Content"

6.1 Overview of the Content Management

Using the Content Manager, you can publish OracleAS Wireless applications to user groups and also manage these user groups. The Content Manager's step-by-step wizards enable you to create the following objects:

Application Links

The Content Manager enables you to publish the master applications as an application link (sometimes referred to as an application). This pointer inherits the parameters of a master application, but can also be used to tailor the core application to the needs of a particular user group or situation. For example, for a master application to deliver restaurant information for an entire city, its adapter takes a single parameter (a location), and returns a list of restaurants throughout the city. While the master application can specify a broad location, such as the city itself, you can create application links based on a specific parameter, such as a district or area within that city. You can then distribute the application links, as appropriate, to user groups that you assemble based on the users' locations.

Folders

Folders enable you to organize application links and bookmarks. When you assign a folder to a user group, you make its subfolders, application links and bookmarks within it accessible to that user group.

Bookmarks

The Content Manager enables you to create a bookmark, a link enabling the user to quickly access an external resource, such as a Web page. In addition to providing the user this shortcut, OracleAS Wireless enables you to create bookmarks that render their content equally well on a variety of devices. End users can set bookmarks in the

OracleAS Wireless Customization Portal. The bookmark appears as a menu selection on the mobile device. OracleAS Wireless does not process the content of the URL target. The format of the target content must be supported by the user's device.

Notifications

A notification (or alert) is an application delivered to users based on trigger conditions, such as time (a user may request that a notification be sent at predetermined times), or based on a change in a condition or value.

The Content Manager enables you to distribute these repository objects to user groups, organize them in a business context appropriate to each user group, and assign them to different categories so that they can only be accessed through specified **access points**.

6.2 Accessing the Content Manager

After you log in to the OracleAS Wireless Tools, you select the Content Manager by clicking the Content tab.

Note: You must be granted either the Super User or Content Manager roles to access the Content Manager.

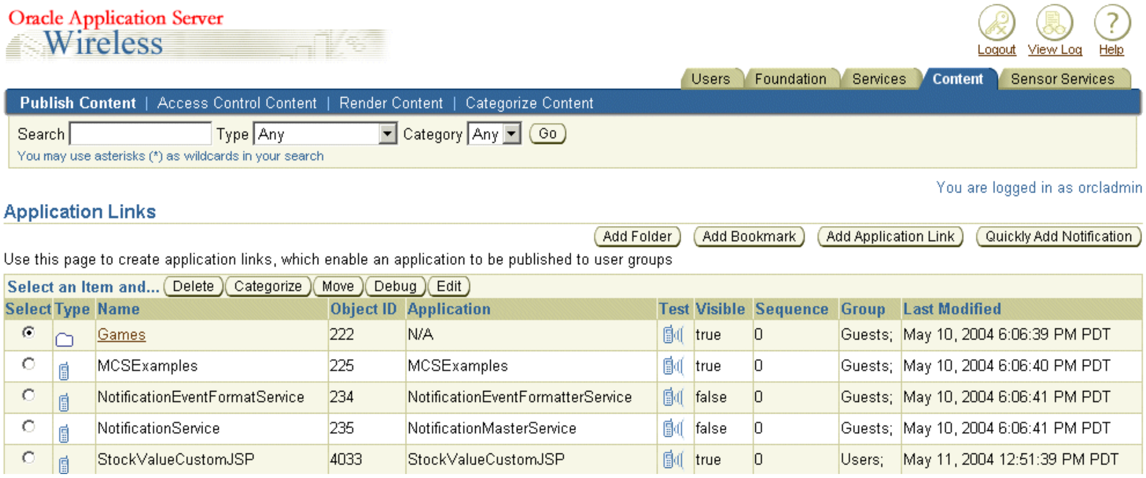
The Content Manager defaults to the *Publish Content* tab (Figure 6–1). The Content Manager organizes the creation, distribution, and publishing of applications into the following tabs (described in Table 6–1).

Table 6–1 Tabs of the Content Manager

Tab	Functions
Publish Content	This tab includes functions to create, edit and delete bookmarks, folders, and application links.
Access Control Content	This tab includes functions to create user groups and assign applications to groups.
Render Content	This tab enables you to group users' home folders by community or by provider.
Categorize Content	This tab enables you to group applications by category. You can also select the access points for these categories.

Each of these tabs includes a browsing page, which enables you to create, edit, or delete an object.

Figure 6–1 The Content Manager



Search [ ] Type Any Category Any

You may use asterisks (*) as wildcards in your search

You are logged in as orcladmin

Application Links

Use this page to create application links, which enable an application to be published to user groups

Select	Type	Name	Object ID	Application	Test	Visible	Sequence	Group	Last Modified
●		Games	222	N/A		true	0	Guests;	May 10, 2004 6:06:39 PM PDT
○		MCSExamples	225	MCSExamples		true	0	Guests;	May 10, 2004 6:06:40 PM PDT
○		NotificationEventFormatService	234	NotificationEventFormatterService		false	0	Guests;	May 10, 2004 6:06:41 PM PDT
○		NotificationService	235	NotificationMasterService		false	0	Guests;	May 10, 2004 6:06:41 PM PDT
○		StockValueCustomJSP	4033	StockValueCustomJSP		true	0	Users;	May 11, 2004 12:51:39 PM PDT

6.3 Managing Application Links

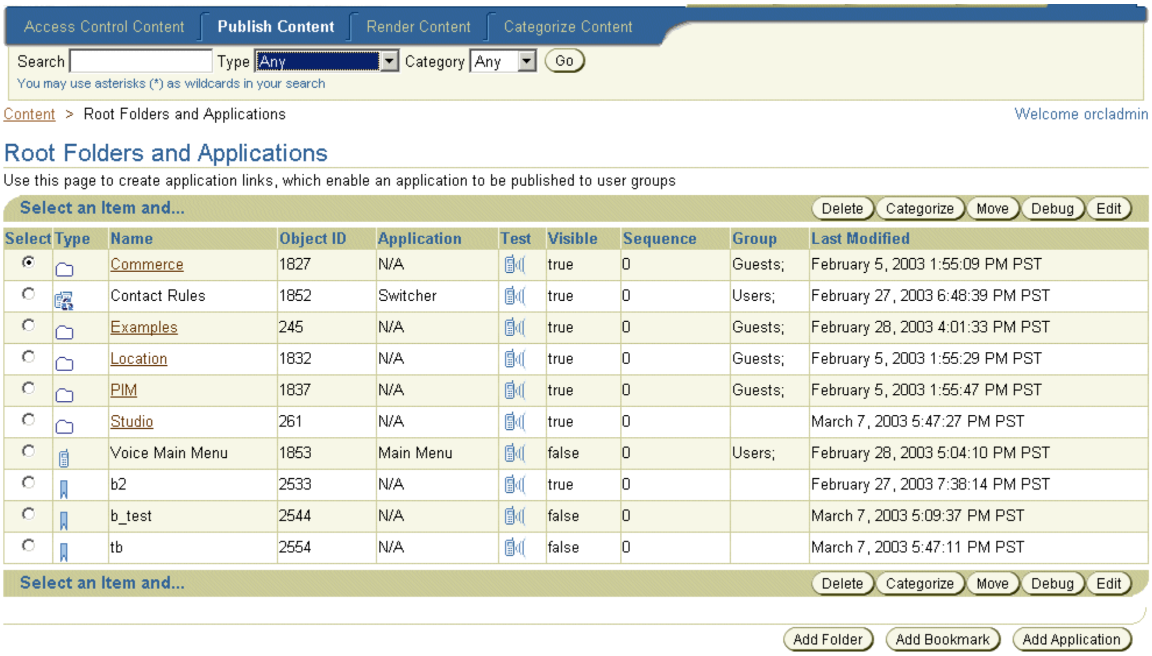
The *Publish Content* tab of the Content Manager enables you to manage application links, bookmarks, and folders.

Clicking the *Publish Content* tab displays the browsing page for application links. When you first access the *Publish Content* tab after logging in, the browsing page displays the folders and applications at the root level (Figure 6–2).

Using this page, you can search for folders, bookmarks, and application links (including asynchronous applications). Clicking the **Add Application** button in this page, you access a wizard that enables you to create an application link based on an existing master application. In addition, this page includes buttons that enable you to add folders and bookmarks. You can also use the browsing page to delete, debug, move, and edit these objects.

For more information on developing multi-channel applications (master applications), see the Oracle Application Server Wireless Developer's Guide.

Figure 6–2 The Browsing Page (From the Publish Content Tab)



Search ______ Type Any Category Any

You may use asterisks (*) as wildcards in your search

Content > Root Folders and Applications

Welcome orcladmin

Root Folders and Applications

Use this page to create application links, which enable an application to be published to user groups

Select an Item and...

Select	Type	Name	Object ID	Application	Test	Visible	Sequence	Group	Last Modified
●	📁	Commerce	1827	N/A	☎	true	0	Guests;	February 5, 2003 1:55:09 PM PST
○	▣	Contact Rules	1852	Switcher	☎	true	0	Users;	February 27, 2003 6:48:39 PM PST
○	📁	Examples	245	N/A	☎	true	0	Guests;	February 28, 2003 4:01:33 PM PST
○	📁	Location	1832	N/A	☎	true	0	Guests;	February 5, 2003 1:55:29 PM PST
○	📁	PIM	1837	N/A	☎	true	0	Guests;	February 5, 2003 1:55:47 PM PST
○	📁	Studio	261	N/A	☎	true	0		March 7, 2003 5:47:27 PM PST
○	📱	Voice Main Menu	1853	Main Menu	☎	false	0	Users;	February 28, 2003 5:04:10 PM PST
○	🔖	b2	2533	N/A	☎	true	0		February 27, 2003 7:38:14 PM PST
○	🔖	b_test	2544	N/A	☎	false	0		March 7, 2003 5:09:37 PM PST
○	🔖	tb	2554	N/A	☎	false	0		March 7, 2003 5:47:11 PM PST

Select an Item and...

6.3.1 Searching for Repository Objects

The browsing page s search function enables you to search for and display the following repository objects:

- Application
- Asynchronous Application
- Bookmark
- Folder

The search field, when used in conjunction with the *Type* and *Category* lists, enable you to narrow your searches to a specific type and category. The results display in the Search Result page, which is described by Table 6–2.

Table 6–2 Elements of the Search Result Section of the Applications Page

Element	Description
Name	The name of the folder or service. Clicking the name of a folder displays its subfolders.
Object ID	The Object ID stored in the database.
Full Path	The route to a repository object, with *Applications* at the root. Each node on the route is displayed as a link. Clicking a link reveals a browse page, displaying the subfolders, applications, and bookmarks organized under the folder. Using this browse page, you can perform such functions as creating and deleting applications, bookmarks, and folders.
Visible	If the column displays *true*, then the object is visible and therefore accessible to an end user. If *false*, then the object is not visible (and the user cannot access it).
Test	Clicking the phone icon enables you to view the application on a simulator.

Table 6–2 (Cont.) Elements of the Search Result Section of the Applications Page

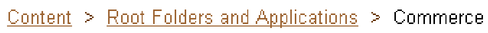
Element	Description
Sequence	The order in which applications and folders appear on output devices. By default, these appear in order by sequence number and then by name. You can enter values in the sequence fields to rearrange the order in which the applications and folder appear. By default, OracleAS Wireless sorts applications and folders in ascending order by sequence number, then by name. See also Section 6.3.2.2.
Group	The user group to which the object is assigned.
Last Modified	The last time the object was modified.

Note: In the Search field, you can find an object by entering a SQL `LIKE` clause pattern matching text (* or %). For example, entering *Per%* in the *Search* field returns all objects beginning with *per*.

6.3.2 Creating a Folder

You can organize your repository objects into a hierarchy by creating subfolders. These subfolders, which can represent topic areas, can be nested into other subfolders. When you create a subfolder, the Content Manager displays it as a link, allowing you to "drill down" or traverse deeper into the hierarchy with each successive click. OracleAS Wireless displays the structure of the hierarchy as a navigation path (Figure 6–3), enabling you to see the level that you currently access and to move back to any previous folder in the hierarchy.

Figure 6–3 The Navigation Path



Content > Root Folders and Applications > Commerce

Creating a folder is a two-step process; you first define the basic parameters for a folder, such as its name, and then you assign the rendering options that dictate the display style for the folder and its contents.

6.3.2.1 Step 1: Defining the Basic Parameters for a Folder

From the browsing page, click **Add Folder.** The *General* page appears, displaying the basic parameters of the folder (described in Table 6–3)

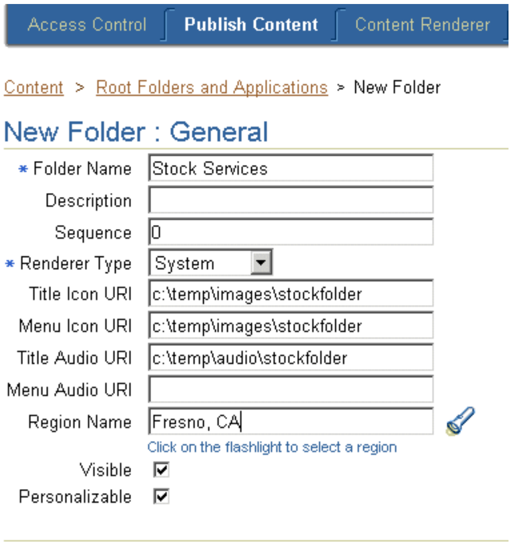
Table 6–3 Parameters of the Content Manager Create Folder Page

Parameter	Value
Folder Name	The name of the folder. This is a required field.
Description	A description of the folder.
Sequence	The order in which applications and folders appear on output devices. By default, these appear in order by sequence number, then name. You can enter values in the sequence fields to rearrange the order in which the applications and folders appear. By default, OracleAS Wireless sorts applications and folders in ascending order by sequence number, then by name.
Language	A drop-down list of display languages for the folder. Any applications or subfolders contained in this folder must have the same display language. Users cannot access these objects if their display language differs from that of the parent folder.

Table 6–3 (Cont.) Parameters of the Content Manager Create Folder Page

Parameter	Value
Renderer Type	A list of the renderer types for a folder. These include: ■ System: The default system object sorting styles. ■ Custom: The object display and sorting styles of another folder or application that dictates the display logic. ■ Inherited: The display style of an ancestor folder which has a custom renderer. If there is no ancestor folder or if the ancestor has a no custom rendering, then the default system sorting style is applied. This is a required field.
Title Icon URI	The URI of an image used as the icon that appears on top of the page when this folder becomes the current folder. You do not need to specify the format type in this URI, as OracleAS Wireless selects the image format appropriate to the user's device.
Menu Icon URI	The URI of an image used as the icon that appears next to the folder in a menu listing. You do not need to specify the format type in this URI, as OracleAS Wireless selects the image format appropriate to the user's device.
Title Audio URI	The URI of the audio file (for example, a .wav file) read aloud by voice-reader software when users access a folder. You do not need to specify the format type in this URI, as OracleAS Wireless selects the audio file format appropriate to the user's device.
Menu Audio URI	The URI of the audio file (for example, a .wav file) read aloud by voice-reader software along with a folder in a menu listing. You do not need to specify the format type in this URI, as OracleAS Wireless selects the audio file format appropriate to the user's device.
Region Name	The area, such as a continent, country, or city, that is associated with the folder. If you assign a region to a folder, then users can only view that folder and its contents when they are in the assigned region.
Visible	Selecting this check box makes the folder visible to the end user. If you do not select this option, then the folder and its contents are neither visible nor accessible to the end user.
Personalizable	Selecting this option enables end users to customize their user views using the OracleAS Wireless Customization or from the device by reordering, hiding, or showing this folder.

Figure 6–4 The Create Folder Page (General Parameters)



Content > Root Folders and Applications > New Folder

New Folder : General

* Folder Name	Stock Services
Description	
Sequence	0
* Renderer Type	System
Title Icon URI	c:\temp\images\stockfolder
Menu Icon URI	c:\temp\images\stockfolder
Title Audio URI	c:\temp\audio\stockfolder
Menu Audio URI	
Region Name	Fresno, CA
	Click on the flashlight to select a region
Visible	☑
Personalizable	☑

6.3.2.2 Step 2: Assigning the Rendering Options

The *Rendering* page displays options specific to the rendering type you selected when setting the basic parameters for the folder.

Selecting the System Rendering Options

If you select **System** as the rendering option, then you can select from among the following sorting options that include the ascending and descending sorting style for folders by:

- ID
- Name
- Last Modified Date
- Sequence Number
- Access Count

By default, folders appear by sequence number, then by name.

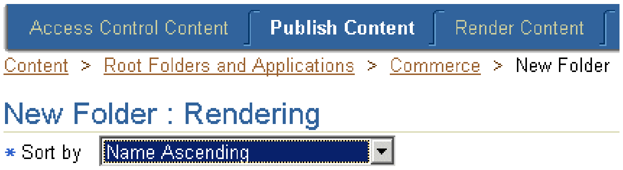
Setting the Customized Rendering Options

If you select the **Custom** rendering options, then you select a folder or application with the appropriate rendering style.

Setting the Inherited Rendering Options

If the folder is not a child of another folder (or if none of its ancestor folders have a customized renderer), then OracleAS Wireless notes the inherited renderer as *N/A* until the folder is moved under a parent folder with a customized renderer. Use the **Move** function to place the folder within a folder configured with the appropriate rendering style. See Section 6.3.10 for information on moving objects.

Figure 6–5 The Folder Rendering Page



Content > Root Folders and Applications > Commerce > New Folder

New Folder : Rendering

* Sort by Name Ascending

6.3.3 Editing a Folder

The **Edit** button enables you to modify the values for a selected folder. After you have modified the appropriate values, click **Apply** to commit your changes. Clicking **Cancel** sets the parameters back to their original values and returns you to the browse page. See Section 6.3.2.1 for information on the basic folder parameters. See Section 6.3.2.2 for information on the folder rendering options.

6.3.4 Creating an Application Link

You create an application link (also known as an application) to publish a master application to users. The master application, which is created using the Service Manager, formss the core of the application link. By clicking **Add Application**, you can create an application link using a wizard which guides you through each step of the creation process, from the first step of basing the application link on an existing master application, to the steps that follow for setting the general information for the application and editing or adding input parameters.

If you base the application link on an asynchronous-enabled application, then you can assign Async Agent properties to the application link, thus enabling customers to access the application using asynchronous messaging technologies such as SMS, e-mail or two-way pagers.

Step 1: Selecting a Master Application

From the *Master Application* page, you can select a master application as the basis for the application link. This master application is the core functionality of the application link.

Step 2: Entering the General Information

The *General* page enables you to set such basic information for the application link, such as the name for the application and the short names for the application (if it is based upon an asynchronous-enabled master application). A **short name** is an easily referenced name for the application entered by end users when accessing applications from asynchronous devices. Use the *Up* and *Down* arrows to prioritize the order in which these short names appear in a help message.

Selecting DRM Policies for a J2ME Application

If this application link is based on a J2ME application, then you can also select a DRM (digital rights management) policy, which controls the digital rights of the J2ME application by defining the user access to the application. For example, a digital rights policy can restrict the user's access to a downloaded application to a certain time period, (as in the case of a trial period), or can limit the number of times a user can download an application. The DRM policies are created by Foundation Developers. For more information, see Section 8.7.

Step 3: Entering New Input Parameters for the Application Link

The *Input Parameters* page enables you to set the input parameters for your application link. The input parameters for the application link are those set for the master application on which you based your application link. You can only change the parameters which the Application Developer designated as *Modifiable*. For more information on creating master applications, see the Oracle Application Server Wireless Developer's Guide.

Table 6–4 describes the input parameters included in the *Input Parameters* page.

Table 6–4 The Input Parameters for an Application Link

Parameter	Value
Name	The name of the input parameter. The *Application Link Creation Wizard* sets the name of the input parameter by querying the definitiion of the master application. This field cannot be edited.
Caption	The label describing this parameter used by OracleAS Wireless when prompting for user input.
Comment	For master applications based on the Web Integration adapter, OracleAS Wireless automatically populates this field with the name of the WIDL service that uses the parameter. For applications based on other adapters, this column documents the parameter. The comment is for internal use only and cannot be edited.

Table 6–4 (Cont.) The Input Parameters for an Application Link

Parameter	Value
Format	This mask sets the expected data entry mode for the user device. For example, if you expect the user to enter numbers for the parameter, you use the format code N. (This works only with WML 1.1-compliant devices.) The default format is *M (which allows the entry of any character). Other formats include: ▪ A, for entry of uppercase letters or punctuation. ▪ a, for entry of lowercase letters or punctuation. ▪ N, for entry of numeric characters. ▪ X, for entry of uppercase letters. ▪ x, for entry of lowercase letters. For a complete list of formats, see the *Wireless Application Protocol Wireless Markup Language Specification, Version 1.1*. This value cannot be edited.
Mandatory	If this check box has been selected, then the parameter must have a value. If the check box is clear, then parameters are optional. This cannot be edited.
Customizable	Specifies whether the end user using the application link can enter values from a mobile device. You can enable most output parameters to be customized by the user.
Value	The default value for the parameter set using the Service Manager. You can override these default values using the Content Manager. If you specify a default value, then OracleAS Wireless does not prompt the user for a value.

Step 4: Assigning the Asynchronous Capabilities to the Application Link.

Use this page to assign the Async Agent capabilities to application. To use this page, you must base the application link on an asynchronous-enabled master application.

Table 6–5 describes the parameters for Async Agents.

Table 6–5 Parameters of the Async Agent Page

Parameter	Value
Async Command Line Syntax Help	The command syntax or usage text. This text is returned to the user when the user issues an application help command to the Async Listener.
Routing Information	Select an item and click **Edit** to access the editing function for the routing presets. For more information, see Section 6.3.4.1.
Application-Specific Address List	The application-specific address to which users send the service invocation messages. Enter this address in the format appropriate to the following device types (SMS or e-mail). For example, enter *stock@oraclemobile.com* as the service address for e-mail. This is an optional parameter.
Async Application Argument List	The default value for each argument. Use the *Move Up* and *Move Down* functions to map the asynchronous application arguments to the input arguments.

6.3.4.1 Editing the Routing Presets

OracleAS Wireless includes a pre-seeded preset, *_MESSAGE_ROUTE_*, whose attributes set the routing information for an asynchronous application.

Routing information, along with application link categories, supports **Premium SMS** and **Reverse Charge SMS**. The routing information enables such information as billing (the Large Account) to be associated with the application, so that the value is returned with the result message. This information is eventually carried over to a PremiumSMS or ReverseCharge operator so that the correct account is charged for the message. For more information on application link categories, see Section 6.7.1.

To edit the routing information, select the routing method (from the *Async Agent Information* page) and then click **Edit**. Enter the values for the routing options as needed. Table 6–6 describes the routing options.

Table 6–6 The Routing Options

Routing Option	Description
CHANNEL	A name of the logical channel through which the message should be sent. This field can be used to store the value of the Large Account field for PremiumSMS.
REVERSE_CHANNEL	The logical channel for the reverse traffic. Both PremiumSMS and ReverseCharge can use this field to store the value of the reply to the Large Account.
COST_LEVEL	The cost level for message delivery. Do not enter a value into this field for PremiumSMS; for ReverseCharge, enter a value that describes the tariff class.

Step 5: Entering the Additional Information for an Application Link

In the final page of the wizard, you define optional parameters for the menu list configuration and user-form submission type. Table 6–7 describes the parameters for this page.

Table 6–7 The Additional Information Parameters for an Application LInk

Parameter	Value
Description	A description of the application link.
Sequence	The order in which application links appear on output devices. By default, these appear in order by sequence number and then by name. You can enter values in the sequence fields to rearrange the order in which the application links appear and then set parent folder renderer type as *System*, and the parent folder sorting option as *Sequence Number.* By default, OracleAS Wireless sorts applications in ascending order by sequence number, then by name. See Section 6.3.2.2 for more information on setting the *System* folder rendering option.
Cost	The invocation cost to the user for accessing the application link. If the cost of the application link is not zero (0), then OracleAS Wireless logs the application link cost invocation in the *tx_panama.log* file.
Language	A drop-down list of display languages for the application link. Users cannot access an application link if their display language differs from that associated with this application link.
Title Icon URI	The URI of an image used as the icon that appears on top of the page when this application link becomes the current application. You do not need to specify the format type in this URI, as OracleAS Wireless selects the image format appropriate to the user's device.
Menu Icon URI	The URI of an image used as the icon that appears next to the application link in a menu listing. You do not need to specify the format type in this URI, as OracleAS Wireless selects the image format appropriate to the user's device.
Title Audio URI	The URI of the audio file (for example, a `.wav` file) read aloud by voice-reader software when users access a service. You do not need to specify the format type in this URI, as OracleAS Wireless selects the audio file format appropriate to the user's device.
Menu Audio URI	The URI of the audio file (for example, a `.wav` file) read aloud by voice-reader software along with the service in a menu listing. You do not need to specify the format type in this URI, as OracleAS Wireless selects the audio file format appropriate to the user's device.

Table 6–7 (Cont.) The Additional Information Parameters for an Application LInk

Parameter	Value
Region Name	The area, such as a continent, country, or city, that is associated with the application. If you assign a region to an application link, then users can only view that application link when they are in the assigned region.
Visible	Select this option to make the application link visible (and therefore accessible) to the end user. If you do not select this option, then end users cannot see (or access) this application link. You can opt not to select this option for application links which are under construction.
Personalizable	Selecting this option enables end users to customize their user views in the Wireless Customization Portal or on the mobile device for reordering, hiding, or showing this application link.

6.3.5 Editing an Application Link

The editing page enables you to change or update the parameter values for a selected application link. To access the editing page, select the application link in the browsing page and then click the **Edit** button. From the editing page's menu, you can select the values that you want to edit, such as those for the general parameters, the input parameters, the Async-agent parameters (if applicable), and the additional parameters (Step 2 through Step 4 of the wizard). You cannot change the master application on which the application link is based.

Note: You can only edit the input parameters of an application link if the input parameters of the master application on which it is based have been designated as *Modifiable*. For more information on developing master applications, see the Oracle Application Server Wireless Developer's Guide.

6.3.5.1 Certifying an Application Link Based on a J2ME Application

When editing an application based on a J2ME (Java 2 Micro Edition) master application, the menu of the editing page includes another option, **API Scan**. This option enables you to select an API scan policy that checks the application for malicious APIs calls which may damage a user's device. These policies are defined using the Foundation Manager. For more information, see Section 8.8.

To scan a policy, select **API Scan** and then the appropriate version of an API scan policy and then click **Certify**.

6.3.6 Testing an Application Link

The Content Manager enables you to test a service and display it on a phone simulator.

To test an application link:

1. From the browsing page, select the application link that you wish to test.
2. Click the telephone icon in the *Run Application* column, which is located in the same row as the selected application link. The phone simulator appears, displaying the application link.

6.3.7 Debugging an Application Link

The Content Manager enables you to simultaneously view an application link on a phone simulator, in OracleAS Wireless XML, or device markup languages.

Transformers, in the form of XSLT style sheets or Java classes, convert the content returned by OracleAS Wireless adapters into the format best suited to a particular platform.

To test a service:

1. On the browsing page, select an application link.
2. Click **Debug**. The *Debug Application Link* page appears.
3. Select from among the following output formats:
   - Adapter XML Result

     Selecting this result type enables you to see OracleAS Wireless source content in the AdapterResult format, the intermediary format between the source and to that of the target output device. Source content in the AdapterResult format must be converted into SimpleResult format before it can be delivered to a target device. If no text displays in the *Result* panel, then no AdapterResult has been produced.
   - OracleAS Wireless XML Result

     Selecting OracleAS Wireless XML Result displays the source content in OracleAS Wireless' SimpleResult format of the output that is returned by an adapter.
   - Device Result

     The *Device Transformer* menu lists the devices in the repository. Selecting a device enables you to see the final markup language for that device.
4. Click **Set Parameters**.
5. Click **Run Application**. The application link appears on a phone simulator. The selected result appears in the Application Result window.

Setting the Display Length of the Logging File

The *System Log* section enables you to set the number of lines from the end of the server's system log file that you want to view.

To set viewing options:

1. Enter the number of lines from the end of the system log that you want to review.
2. Click **Refresh Log**. The specified number of lines from the end of the system log appear.

6.3.8 Creating User Bookmarks

The Content Manager enables you to create a bookmark, a link enabling the user to quickly access an external resource, such as a Web page. In addition to providing the user a this shortcut, OracleAS Wireless enables you to create bookmarks that render their content on a variety of devices.

With OracleAS Wireless, a bookmark displays equally well on all of the different devices registered to a OracleAS Wireless user, because you can associate multiple URLs with a single bookmark. Each of these URLs supplies the markup suitable to the content type supported by the requesting device.

For example, you create a bookmark, *myoracleBK*, which has the following two URLs:

- `www.oracle.com` with the text/html MIME type
- `wap.oracle.com` with the text/hdml MIME type

Logging in through a desktop browser, a user sees *myoracleBK*. Clicking this bookmark reveals the page *www.oracle.com*.

A user logging in from a device supporting the text/hdml MIME type also sees *myoracleBK*, but clicking this bookmark reveals the page *wap.oracle.com*

Clicking **Add Bookmark** in the browsing page invokes the *New Bookmark* page, which includes the parameters described in Table 6–8.

Table 6–8 Parameters of the New Bookmark Page

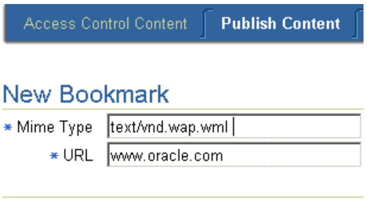
Parameter	Value
Bookmark Name	The name of the bookmark. This is a required field.
Description	A description of the bookmark.
Sequence	The order, as specified by an integer value, in which the bookmarks appear on output devices. By default, these appear in order by sequence number and then by name.
Cost	The cost to the user for accessing the bookmark.
Region Name	The area, such as a continent, country, or city, that is associated with the bookmark. If you assign a region to a bookmark, then users can only view that bookmark and its contents when they are in the assigned region.
Visible	Selecting this check box makes the bookmark visible to the end user. Leaving this check box clear prevents end users from seeing or accessing the bookmark.
Personalizable	Selecting this option enables end users to customize their user views in the OracleAS Wireless Customization Portal or on the mobile device by reordering or hiding and showing bookmarks.

In addition to these parameters, (whose values define the basic settings for the bookmark), the *New Bookmark* page also includes a table listing URLs and MIME types that you associate with this bookmark. This table also notes the default MIME type, which you can set by selecting a MIME type and then by clicking **Set Default**.

> **Note:** Only the URL for the `text/vnd.wap.wml` MIME type can be set as the default. For example, a user cannot access a bookmark with both a `text/vnd.wap.wml URL` and a voice URL through a PC unless you click **Set Default** to designate the `text/vnd.wap.wml` URL as the default URL. Because OracleAS Wireless only supports transcoding through `text/vnd.wap.wml`, the bookmark appears on the PC, as OracleAS Wireless transforms the WML result into SimpleResult and then into HTML. There can be only one URL with the text/vnd.wap.wml type. You do not need to select the `text/vnd.wap.wml` URL when you click **Set Default**.

You can add other URLs or MIME types to the table by first clicking the **Add** button and then by defining the values for URL and MIME type in the following page (Figure 6–6.)

Figure 6–6 Adding a New MIME Type



New Bookmark

* Mime Type: text/vnd.wap.wml

* URL: www.oracle.com

6.3.9 Editing a Bookmark

The *Edit Bookmark* page, which enables you to alter a bookmark, which you access by selecting a bookmark and then by clicking *Edit*.

6.3.10 Moving Folders, Application Links, and Bookmarks

You can organize application links, folders, and bookmarks in a business context appropriate to a user group by using the Content Manager's **Move** function.

To move application links, folders, or bookmarks:

1. From the browsing page, select the folder, application link, or bookmark that you want to move.
2. Click **Move**.
3. Select the new folder for the object. If necessary, click the folder to drill down to the appropriate subfolder. OracleAS Wireless tracks your position in the hierarchy through the navigation path. For more information on the navigation path, see Section 6.3.2.
4. Click **Move Here**. The Content Manager displays the selected object in its new folder.

6.4 Managing Notifications

The **Quickly Add Notifcation** button enables you to create a notification.

6.4.1 Creating a Notification

By clicking the **Quickly Create Notification** button in the browsing page, you create a notification using a wizard that provides a separate page for each step of the creation process.

6.4.1.1 Step 1: Entering the Basic Configuration Parameters for the Notification

Define the following configuration parameters for the notification in the *Basic Info.* page, the first page in the notification creation wizard. Table 6–9 describes the parameters of the *Basic Info*. page.

Table 6–9 Basic Configuration Parameters for a Notification

Parameter	Value
Name	The name of the notification. This is a required parameter.
Description	A description of the notification.
Subscriber Filtering Hook	A Java class name. This hook enables you to filter out subscribers to the qualified notifications before these notifications are sent to the messaging server.

Table 6–9 (Cont.) Basic Configuration Parameters for a Notification

Parameter	Value
Value-Based	Specifies whether this notification triggers upon the receipt of an event.
Data Feeder	A list of data feed sources. If this notification is value-based, then the value entered in this field must point to a data feeder.
Location-Based Enabled	Specifies whether this notification triggers upon verification of location conditions.
Time-Enabled	Specifies whether this notification triggers at predetermined times. The frequency options are daily, week day, and weekend. The user profile provides the time zone information.

Click **Next**. The *Trigger Conditions* page appears.

6.4.1.2 Step 2: Setting the Trigger Conditions for the Notification

The *Trigger Conditions* page enables you to set the conditions that invoke a notification on end users' devices. For example, if you create a notification that alerts users of a stock price, you set the conditions that allow an end user to request a notification when the stock has risen above, or fallen below, a certain price.

Table 6–10 describes the parameters of the *Trigger Conditions* page.

Table 6–10 Trigger Conditions for Notifications

Parameter	Value
Condition Name	The name of the alert trigger for the notification. The trigger name, which is limited to 30 characters, must contain only alphanumeric characters and an underscore. In addition, the trigger name cannot start with a numeric character and cannot use SQL reserved words. End users see this label when they subscribe to a notification application.
Trigger Parameter	The trigger parameter is an element in a data feeder that you define a trigger condition against. For example, if a data feeder for a stock alert service includes an output parameter of *stock price*, you could select *stock price* as the trigger parameters for a condition name. For information on setting the output parameters of a data feeder, see Editing the Output Parameters of a Data Feeder in the Developer's Guide.
Condition Type	The condition, in relation to the value set by the end user, which triggers the notification.
Default Value	The default value for the parameter. If you specify a default value, OracleAS Wireless does not prompt the user for a value. Default values can be overridden by a value specified by an application created by the Content Manager or, if the parameter is visible to the user, by the user through OracleAS Wireless Customization.

Setting the Relationship Between Trigger Conditions

Select an *AND* relationship (both conditions must be met) or an *OR* relationship (any of the conditions must be met).

Selecting a Trigger Condition

To select a trigger condition:

1. From the list of trigger conditions, select the trigger condition.
2. Edit the *Condition Type*, *Trigger Parameter*, or *Default Value* fields as needed.
3. Click **Apply**.

Adding a New Trigger Condition

To add a new Trigger Condition

1. Enter the name for the trigger condition in the *Condition* field.
2. Enter text used for prompting input from end users in the *Caption* field.
3. Select a trigger parameter from the drop-down list in the *Trigger Parameter* field.
4. Select a condition type from the drop-down list in the *Condition Type* field. Condition types depend on the data type of the trigger parameter.

   If the data type is a number, then the conditions include:

- Less Than
- Greater Than
- Equal
- Less Than and Equal
- Greater Than and Equal
- Less Than Absolute Value
- Greater Than Absolute Value
- Equal Absolute Value
- Less Than and Equal Absolute Value
- Greater Than and Equal Absolute Value
- Value Change (The condition value for this type can only be *0* or *1*, where *0* means *no trigger* and *1* means *trigger when value changes*. The default value is *0*.)

If the data type is text, then the condition types include:

- Exact Match
- Not Match
- Contain
- Not Contain
- Begin With
- End With
- Value Change (The condition value for this type can only be *0* or *1*, where *0* means *no trigger* and *1* means *trigger when value changes*. The default value is *0*.)

1. Enter a default value for the trigger condition in the Default Value field.
2. Click **Add**.
3. Click **Next**. The *Message Template* page appears.

6.4.1.3 Step 3: Creating the Message Template

The *Message Template* page enables you to create a message template by entering SimpleText stylesheet. In this stylesheet, the data feeder output values are the dynamic values. The stylesheet illustrated in Example 6–1 represents these values as *sym*, *price* and *change* **.**

Example 6–1 SimpleText StyleSheet Used for a Message Template

```
<SimpleResult>
   <SimpleContainer>
      <SimpleText>
      <SimpleTitle>OracleAS Wireless</SimpleTitle>
      <SimpleTextItem>Notification with price: $price; and change: $change: for 
stock: &sym;</SimpleTextItem>
      </SimpleText>
    </SimpleContainer>
</SimpleResult>
```

> **Note:** OracleAS Wireless does not commit any of the values that you have entered until you complete the entire wizard.

6.4.1.4 Step 4: Adding the URLs and Other Information

If this notification has a message template, then select the Use Message Template from Notification option for this notification to use the default messaging URL.

> **Note:** Adding input parameters while creating a notification-enabled application is optional. However, if you select the **Default URL** option (that is, the notification has a message template), then OracleAS Wireless ignores any input parameters created using this tool.

- Select **Use the default messaging URL for the notifications with message template** to use the default messaging generation mechanism. OracleAS Wireless provides a default application JSP to generate the notification message based on the notification template. If you select this option along with a notification that includes a message template, then the default JSP handles the message generation. You do not need to provide the URL to the mobile application.
- Select **Generate Non-Personalized Shared Content** so that the mobile application is invoked only once for each incoming event. In this case, the user information passed to the mobile application is the system user, and generated content is shared by the users who are qualified for this event. The mobile application cannot perform any special processing for each user and can not generate personalized message for each user. However, selecting this option improves performance, since the mobile application is invoked only once for multiple notifications.

Click **Finish** to complete the notification.

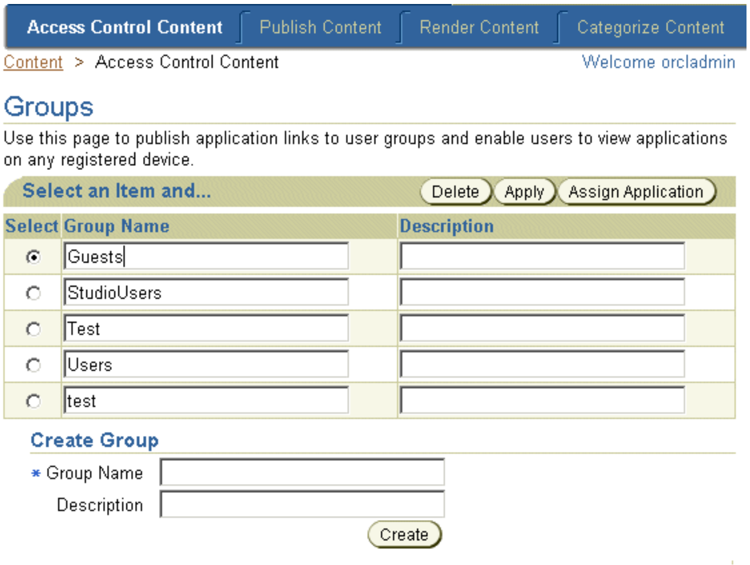
6.5 Defining Access Control

The Content Manager enables you to create, edit, and delete user groups. You can also publish application links to users by assigning them to user groups. When an object, such as a folder, has been published to a user group, an end user belonging to that group can access the object from any device registered with OracleAS Wireless. In addition to creating user groups and assigning objects to them, you can also remove objects from user groups.

6.5.1 Managing a User Group

Clicking the *Access Control Content* tab invokes the *Groups* page (Figure 6–7), which includes a table listing the current user groups. From this table, you can select a user group (using the *Select* button) and then edit it, delete it, or manage the objects assigned to it.

Figure 6–7 The Groups Page



Groups

Use this page to publish application links to user groups and enable users to view applications on any registered device.

Select an Item and...

Select	Group Name	Description
●	Guests	
○	StudioUsers	
○	Test	
○	Users	
○	test	

Create Group

* Group Name	
Description	

Table 6–11 describes the fields and functions of the Groups page.

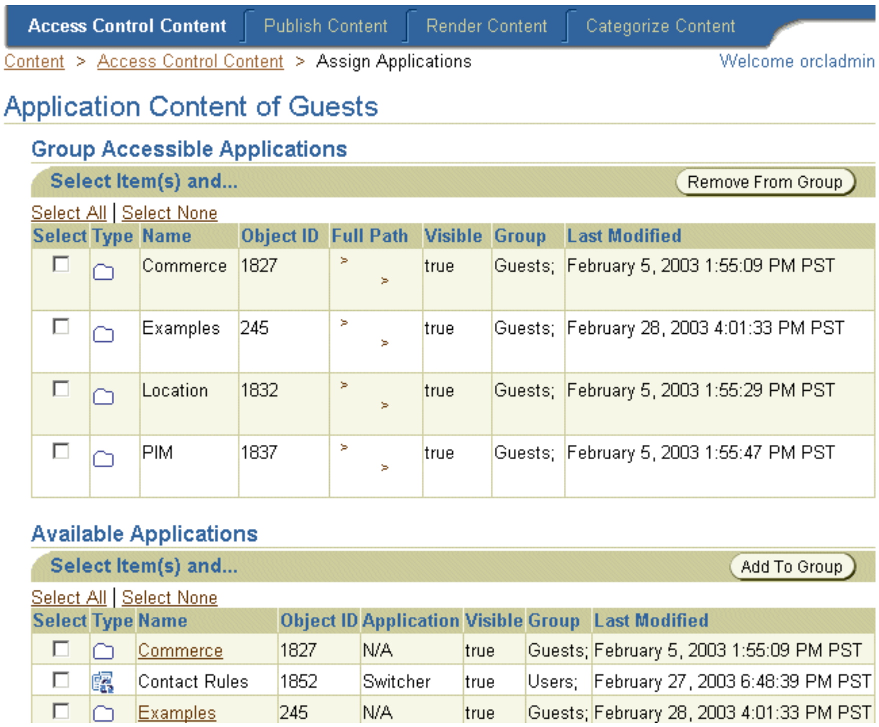
Table 6–11 Elements of the Groups Page

Element	Description
Delete	You can delete a group by selecting it from the table and then by clicking **Delete**.
Apply	After you edit the name or description of a selected group in the table, click **Apply** to save your changes.
Assign Application	Selecting a group and then clicking this button invokes the *Application Content* page, which enables you to manage the objects assigned to the selected group.
Group Name	The name for the user group. This is a required field.
Description	An optional description of the user group.
Create	Enables you to create a user group. The new user group appears in the table, where it can then be selected for editing, deleting, or for content management.

6.5.2 Managing the Contents of a User Group

To manage the contents of a user group, select the group and then click **Assign Applications**. The *Application Content* page for the selected groups appears (Figure 6–8, displaying the objects currently associated with the groups as well as the objects which can be assigned to the group. From this page, you can assign selected application links, bookmarks, or alerts (notifications) to a user group, or remove them from a user group by clicking either the **Add to Group** or **Remove from Group** buttons. Clicking **Finish** saves the changes made to the contents of a user group.

Figure 6–8 The Application Content Page



Content > Access Control Content > Assign Applications

Application Content of Guests

Group Accessible Applications

Select Item(s) and...

Select All | Select None

Select	Type	Name	Object ID	Full Path	Visible	Group	Last Modified
☐	📁	Commerce	1827	> >	true	Guests;	February 5, 2003 1:55:09 PM PST
☐	📁	Examples	245	> >	true	Guests;	February 28, 2003 4:01:33 PM PST
☐	📁	Location	1832	> >	true	Guests;	February 5, 2003 1:55:29 PM PST
☐	📁	PIM	1837	> >	true	Guests;	February 5, 2003 1:55:47 PM PST

Available Applications

Select Item(s) and...

Select All | Select None

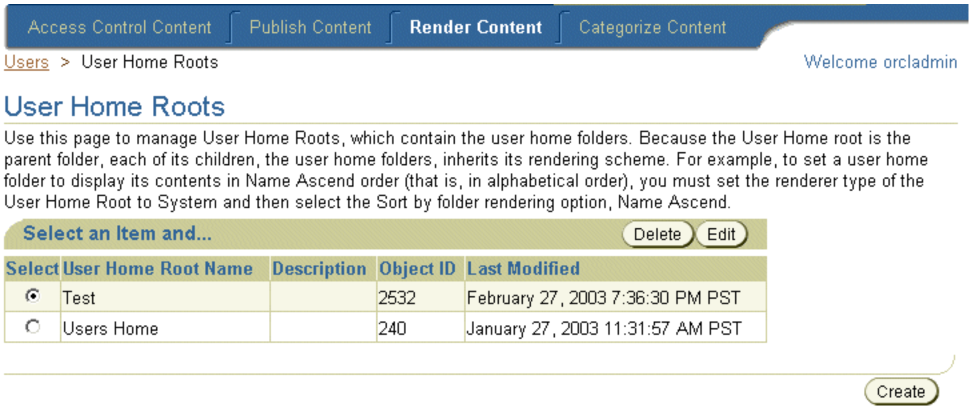
Select	Type	Name	Object ID	Application	Visible	Group	Last Modified
☐	📁	Commerce	1827	N/A	true	Guests;	February 5, 2003 1:55:09 PM PST
☐	▦	Contact Rules	1852	Switcher	true	Users;	February 27, 2003 6:48:39 PM PST
☐	📁	Examples	245	N/A	true	Guests;	February 28, 2003 4:01:33 PM PST

6.6 Creating User Home Root Folders

The *Render Content* tab enables you to group user home folders by user community or by provider. When a user is assigned to a user home root folder, that user's home folder becomes the child of the user home root folder by being placed within it. In addition, user home folders inherit the folder rendering style, or display properties, of their user home root folder. For more information on assigning a user home folder, see Section 5.5.

Selecting the *Render Content* tab displays *User Home Roots* page (Figure 6–9), which includes a table listing the current root folders by name, description, object ID in the database, and by the date that the folder was last modified. From this table, you can both edit and delete selected user home root folders.

Figure 6–9 The User Home Roots Page



Users > User Home Roots

User Home Roots

Use this page to manage User Home Roots, which contain the user home folders. Because the User Home root is the parent folder, each of its children, the user home folders, inherits its rendering scheme. For example, to set a user home folder to display its contents in Name Ascend order (that is, in alphabetical order), you must set the renderer type of the User Home Root to System and then select the Sort by folder rendering option, Name Ascend.

Select an Item and...

Select	User Home Root Name	Description	Object ID	Last Modified
●	Test		2532	February 27, 2003 7:36:30 PM PST
○	Users Home		240	January 27, 2003 11:31:57 AM PST

Clicking the **Create** button enables you to add a new user home root folder. Creating a user home root folder is a two-step process.

Step 1: Entering the General Information

After you click **Create**, the *Genera*l page appears. Table 6–12 described the parameters of this page.

Table 6–12 Parameters of the General Page for User Home Root Folders

Parameter	Value
User Home Root Name	The name of the user home root folder. This is a required field.
Description	A description of the folder.
Renderer Type	A list of the renderer types for a folder. This is a required field. These include: ■ System: The default system object sorting styles. ■ Custom: The object display and sorting styles of another folder or service that dictates the display logic. ■ Inherited: The display style of an ancestor folder which has a custom renderer. If there is no ancestor folder or if the ancestor has a no custom rendering, then the default system sorting style is applied.
Title Icon URI	The URI of an image used as the icon that appears on top of the page when this folder becomes the current folder. You do not need to specify the format type in this URI, as OracleAS Wireless selects the image format appropriate to the user's device.
Menu Icon URI	The URI of an image used as the icon that appears next to the folder in a menu listing. You do not need to specify the format type in this URI, as OracleAS Wireless selects the image format appropriate to the user's device.
Title Audio URI	The URI of the audio file (for example, a `.wav` file) read aloud by voice-reader software when users access a service. You do not need to specify the format type in this URI, as OracleAS Wireless selects the audio file format appropriate to the user's device.
Menu Audio URI	The URI of the audio file (for example, a `.wav` file) read aloud by voice-reader software along with the service in a menu listing. You do not need to specify the format type in this URI, as OracleAS Wireless selects the audio file format appropriate to the user's device.

Step 2: Assigning the Rendering Options

Clicking **Continue** on the General page invokes the second (and final) page used to create a user home root folder, the Rendering page. This page contains the display options specific to the renderer type (*System*, *Inherited*, or *Customized*) that you selected when setting the basic parameters for the user home root folder. Because user home folders are the children of the user home root folders, each user home folder inherits the rendering style of its parent, the user home root folder.

Setting the System Default Rendering Options

If you selected a *System* renderer type, then you can select from among the following sorting options in the *Rendering* page.These options include the ascending and descending sorting style for folders by:

- ID

- Name
- Last Modified Date
- Sequence Number
- Access Count

By default, folders appear by sequence number and then by name. Click **Finish** to complete the user home root folder.

Setting the Customized Rendering Options

If you select the *Custom* renderer, then the *Rendering* page displays the root-level folders and applications. Using the **Select** button, you choose the appropriate folder or application with the appropriate rendering style and then click **Finish** to complete the user home root folder.

Setting the Inherited Rendering Options

If you selected the Inherited renderer option, then click **Finish** in the *Rendering* page. The inherited rendering for a user home root folder is the system default rendering.

6.6.1 Editing a User Home Root Folder

You can edit both the general parameters and the rendering options for a selected user home root folder. To do this, select a folder from the table in the User Home Roots page and then click **Edit**. The editing page appears and defaults to the general parameters set for the selected user home root folder. If you wish to edit the rendering options, select **Rendering** from the menu. Click **Apply** to save your changes. Clicking **Cancel** sets them back to their previous values.

6.6.2 Deleting a User Home Root Folder

You can delete a user home root folder by first selecting from the table in the *User Home Roots* page and then by clicking **Delete**.

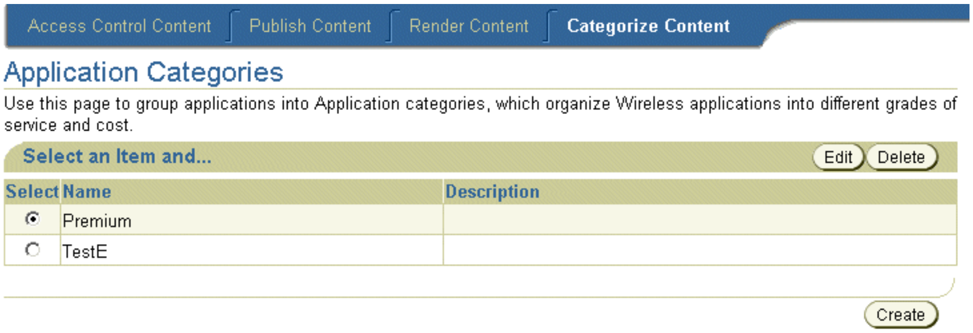
Note: You cannot delete a user home root folder if it contains any user home folders; you must delete all user home folders from a user home root folder before you can delete it.

6.7 Categorizing Content

To support **Premium SMS** and **Reverse Charge SMS**, the Content Manager enables you to create application link categories, which are sets of similar applications. For example, in PremiumSMS, each set of applications having the same premium level can be put into an application link category.

Each access point (for example, an Async address) can be optionally associated with one or more application link categories. Only the associated access points can gain access to applications assigned to a category.

Figure 6–10 The Categorize Content Page



Application Categories

Use this page to group applications into Application categories, which organize Wireless applications into different grades of service and cost.

Select	Name	Description
●	Premium	
○	TestE	

Clicking the *Categorize Content* tab invokes the *Application Categories* page (Figure 6–10), which includes a table listing the current application link categories. From this table, you select an application link category for editing or for deleting. Clicking the **Create** button invokes the *Create Application Category* page, which enables you to create an application category and assign access points to the category. After you create a category, you then associate application links with that category.

6.7.1 Creating an Application Link Category

In the *Create Application Category* page, enter a name for the application link category. For example, enter *Premium*. This is a required field. If needed, enter a description. You then associate an access point with the category, thereby making all of the applications associated with this category accessible through the selected access point. (The access points are created using the System Manager. For more information, see Section 6.7.1.1). If needed, click **Add** to select additional access points.

6.7.1.1 Creating Access Points using the System Manager

Access points, which display in the Content Manager, are created using the System Manager as part of the configuration of the Async Listener. A user having System Administrator privileges sets the values for an access points, which includes a name, delivery type, or site address or number. For the latter value, the address should be the Large Account provided by the Premium SMS operator. For more information on configuring the Async Listener, see Section 3.10.2.1.

For an access point to display in the Content Manager (and in turn, be selected for an application category), the System Administrator cannot select the option, **Allowed to Access All Applications**. If this flag is set, then this access point cannot be associated with a specific application category, because users sending requests to this access point can access all applications, not just those grouped into any one application category.

6.7.2 Assigning Applications to an Application Link Category

After you create an application category, you add asynchronous applications to the application category. To do this, select an application from the browsing page of the Publish Content tab and then click **Categorize**. In the following page, use the **Move** arrows (> and >>) to move an application from the **All Application Categories** pane to the **Associated Application Categories** pane. To remove an application link from an application link category, use the **Remove** arrows (< and <<) to move the selected application link category from the *Associated Application Category* pane to the *All Application Categories* pane.

6.7.3 Adding SMS Routing Information

You can add SMS routing information when creating (or editing) Premium SMS-enabled application links. For example, you can assign the value of the Large Account, to which the reply message should be charged, to the *Channel* field. For more information, see Section 6.3.4.

6.7.3.1 Editing the Routing Definitions

Optionally, you can edit the pre-seeded _MESSAGE_ROUTE_ preset definition so that each portal can customize the message headers which are sent to the SMS driver as the billing information for the result message. For example, you can change the description of ROUTE_COST_LEVEL from *cost level* to *tariff class*, or add or delete meta fields.

By default, the values of the two fields, ROUTE_CHANNEL and ROUTE_REV_CHANNEL, are set to the *From* and *ReplyTo* fields, respectively, of the result message. Because of this, a custom-built driver is not needed to pass information to the Premium SMS operator. To change these mappings, a System Administrator modifies the following attributes in the `system.properties` file:

- `wireless.async.routeinfo.to`
- `wireless.async.routeinfo.replyto.`

7 Administering Mobile Studio

This chapter describes the administration of Mobile Studio. Sections include:

- Section 7.1, "Overview of Mobile Studio"
- Section 7.2, "Configuring Mobile Studio"
- Section 7.3, "Accessing Mobile Studio Administration"
- Section 7.4, "Managing Locales"
- Section 7.5, "Managing Sample Services"
- Section 7.6, "Viewing Statistics"

7.1 Overview of Mobile Studio

Mobile Studio is the online, hosted environment for developing, testing and deploying mobile applications for the OracleAS Wireless platform. Mobile Studio also serves as a Web portal, supporting the wireless developer community in the enterprise and on the Internet.

Because Mobile Studio provides a Web-based interface for the configuration, testing and deployment of wireless applications, developers do not need to download or install anything on their workstations; they need only a Web browser and access to Mobile Studio. Once an application is registered with Mobile Studio, developers can test it using any mobile device or simulator (including voice) and instantly access real-time logs to troubleshoot any issues.

Application providers can easily brand Mobile Studio by customizing its look-and-feel as well as its content and integrate it with their existing Web site.

Mobile Studio can serve as both an interactive development tool and as a one-stop shop for up-to-date information and collateral on the OracleAS Wireless server platform. Mobile Studio extends OracleAS Wireless so that all Mobile Studio accounts are also OracleAS Wireless accounts (and OracleAS Wireless accounts are also Mobile Studio accounts).

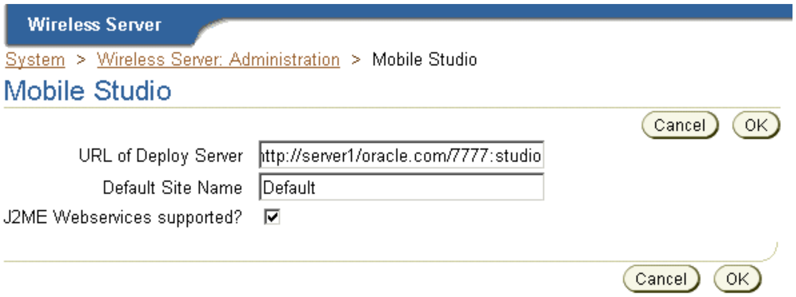
7.2 Configuring Mobile Studio

You can use the System Manager to configure Mobile Studio. To access the configuration page:

1. Click **Wireless Server Administration**. The administration pages appears for the OracleAS Wireless site (Figure 7–1).
2. Select **Mobile Studio** (located in the *General Configuration* section).

For more information on administering the OracleAS Wireless site, see Section 3.9.

Figure 7–1 Mobile Studio Configuration Page



URL of Deploy Server: http://server1/oracle.com/7777:studio

Default Site Name: Default

J2ME Webservices supported? ☑

The Mobile Studio configuration page (Figure 7–1) includes the following parameters, which are described in Table 7–1.

Table 7–1 Parameters of Mobile Studio Configuration Page

Parameter	Value
URL of Deploy Server	The URL of the OracleAS Wireless production instance. Applications created by developers in Mobile Studio (referred to as the development instance) are deployed to this URL. For example, enter `http://myserver.mycompany.com:myport/studio`. If you do not enter the URL in this field, then deployment is disabled.
Default Site Name	The name of the branding (that is, the look-and-feel) which is used as the default. This is pre-seeded with the value *Default*. Application providers can both brand Mobile Studio by customizing its appearance and content and integrate it with an existing Web site. To substitute a branding other than *Default*, enter the name of another branding in this field. For more information on branding, refer to the *Oracle Application Server Wireless Developer's Guide*.
J2ME Webservices Supported?	Select this option to enable the Web services feature of Mobile Studio. By default this option is not selected (that is, the flag is set to *false*). By selecting this option, Mobile Studio's interface displays an additional tab that includes functions that enable developers to register the Web services called from J2ME MIDlets.

After you define Mobile Studio configuration parameters, click *OK*.

> **Note:** You must restart the OracleAS Wireless server for Mobile Studio configuration settings to take effect.

7.3 Accessing Mobile Studio Administration

Access Mobile Studio main page at the following URL:

```
http://<studio_server>:<studio_port>/studio/admin
```

where <*studio_server*> and <*studio_port*> are the name of the host and port number running Mobile Studio instance. These are configured in the Oracle Installer.

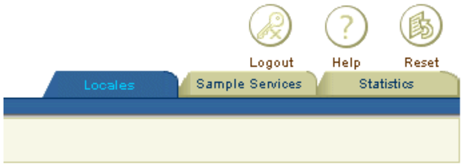
> **Note:** Mobile Studio has been optimized for the latest versions of the Netscape and Internet Explorer browsers.
>
> Mobile Studio is not certified for the older versions of Netscape 4.x or Internet Explorer 4.x.

Enter your Administrator login information as follows:

1. Enter your user name (for example, *orcladmin*).
2. Enter a password (for example, *manager*).
3. Click **Login.** If you entered your login information correctly, then the administration pages appear.

For any of the changes that an administrator makes through the administration pages to be visible to end-users, you must click the **Reset** button (Figure 7–2), which is located on the top right-hand side of the administration pages.

Figure 7–2 The Reset Button

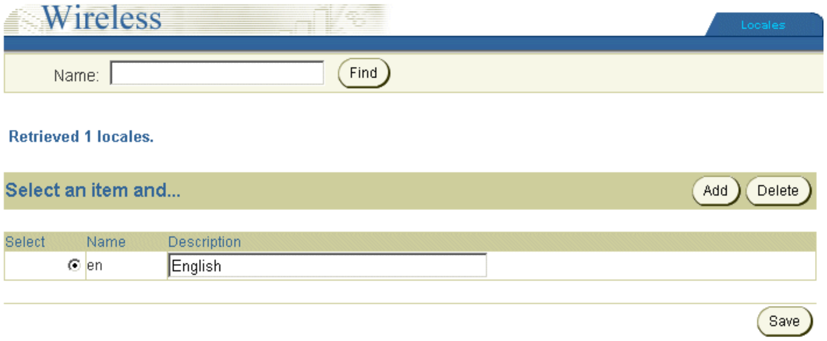


7.4 Managing Locales

The *Locales* page of Mobile Studio (Figure 7–3) displays the enabled locales used for branding Mobile Studio. If you have made no changes to the locales, then only the default enabled locale (English) displays on this page.

This page enables you to find a locale (or locales) using a pattern. You can also add, edit, and delete locales.

Figure 7–3 The Locales Page



Retrieved 1 locales.

Select	Name	Description
●	en	English

7.4.1 Finding a Locale

To find a locale, enter the name or a pattern for the locale in the *Name* field, and then click the **Find** button. A list of the locales matching the name or pattern appears.

7.4.2 Adding a Locale

To add a locale, click the **Add** button. Mobile Studio adds a new row to the list of locales with empty *Name* and *Description* fields. To create this locale, you must enter values into these fields. For example, enter *ru* in the *Name* field and *Russian* in the *Description* field. Click *Save* to commit (store) the values.

7.4.3 Editing a Locale

You can edit a locale by modifying the name and description values. After you have changed the appropriate value (or values), click **Save** to commit the changes.

Note: You cannot edit the name of a default locale.

7.4.4 Deleting a Locale

To delete a locale, select the locale and then click the **Delete** button. To commit the deletion, click the **Save** button. To undo the deletion, click the **Undelete** button. The **Undelete** button appears if you have just deleted any configuration parameters, but have not yet saved your changes.

Note: You cannot delete a default locale.

Note Also: The change and deletions are not committed until you click the *Save* button.

7.4.5 Enabling the Default Locales

Mobile Studio ships with default bundles for 28 different locales (listed in table Table 7–2).

Table 7–2 The Default Locale Bundles for Mobile Studio

Name	Description	Name	Description
ar	Arabic	ko	Korean
cs	Czech	nl	Dutch
da	Danish	no	Norwegian
de	German	pl	Polish
el	Greek	pt	Portuguese
es	Spanish	Pt_BR	Portuguese (Brazil)
es_ES	Spanish (Spain)	ro	Romanian
fi	Finnish	ru	Russian
fr	French	sk	Slovak
fr_CA	French (Canada)	sv	Swedish
hu	Hungarian	th	Thai
it	Italian	tr	Turkish
iw	Hebrew	Zh_CN	Chinese (PRC)
ja	Japanese	Zh_TW	Chinese (Taiwan)

You enable these locales after you have entered them as described in Section 7.4.2 and reset the system by clicking the **Reset** button. For example, to support users whose preferred locale is *ru*, you add *ru* and then the locale's description (for example: *Russian*), and then click the **Reset** button to enable the locale for users.

7.4.5.1 Adding New Locales

If you want to support a locale that is not among those listed in Table 7–2, or if you want to add a locale to your own branding, then you may have to create additional supporting resources, such as text translations. For more information, see the Oracle Application Server Wireless Developer's Guide.

7.4.6 Resolving Locales

The following is a description of the algorithm used by Mobile Studio to resolve which locale to use, given the list of preferred locales for the user, which can be obtained from the request:

1. Mobile Studio searches for the preferred locale (*L*, for example) in the list of enabled locales for Mobile Studio. If Mobile Studio finds *L*, then Mobile Studio returns it and stops the search. If *L* cannot be found, then Mobile Studio performs another search on a new *L* by using only the language part of *L*. For example, if *en_US* cannot be found, then Mobile Studio searches only for *en* instead. If the second search succeeds, then Mobile Studio returns *en* and stops the search.
2. If Mobile Studio finished the search without finding the locale, then Mobile Studio returns the default locale of the default site (if that default is enabled).
3. If after Step 2, Mobile Studio still cannot find the preferred locale, then it returns the locale *en*.

Adding Additional Locales

These instructions illustrate how to add Hindi (*hi*) as a locale. To add a locale:

1. Provide a `DefaultSite_hi.properties` file (or use Mobile Studio's resource administration pages to provide a value of locale hi for each of the resources that must be changed).

   To add the file to the application:

   a. From the OracleAS Wireless root directory, navigate to `iaswv20/wireless/lib`, and find the `studio.jar` file.

   b. Unjar the file and add `DefaultSite_hi.properties` to the extracted files.

   c. Jar all the files back into `studio.jar`.

2. Provide the `messages_hi.properties` file for messages.

   To add the file to the application:

   a. From the OracleAS Wireless root directory, navigate to `iaswv20/wireless/server/classes/messages/oracle/panama/studio.`

   b. `Insert messages_hi.properties` into that directory.

3. Provide the `ommsg_hi.js` file for javascript messages.

   To add the file to the application, follow these steps:

   a. From the OracleAS Wireless root directory, navigate to `iaswv20/wireless/j2ee/applications/studio/studio-web/javascript/`.

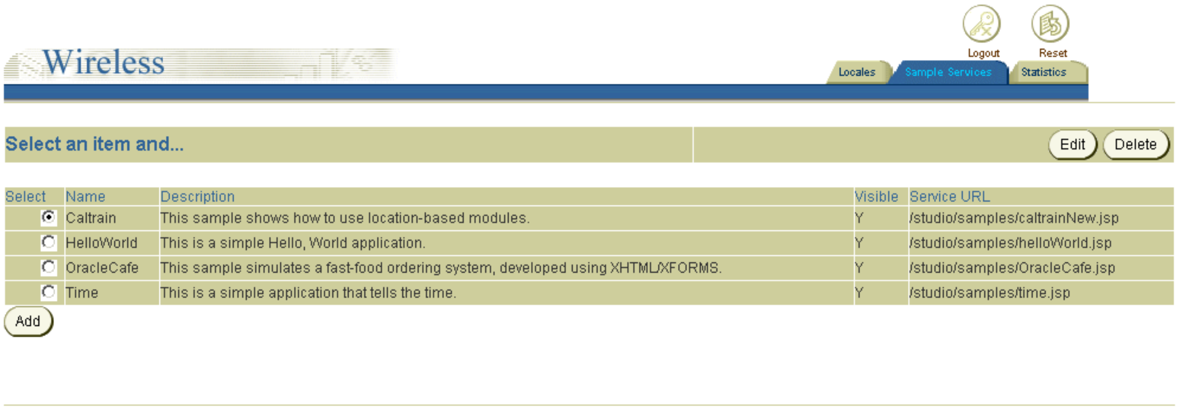
   b. Insert `ommsg_hi.js` in that directory.

4. Restart the instance after making these changes.

7.5 Managing Sample Services

The *Sample Services* page (Figure 7–4) of Mobile Studio enables you to manage the sample services (that is, the applications) that are available in Mobile studio. These services include *Caltrain*, *HelloWorld*, *OracleCafe*, and *Time*. From this page, you can add, edit, and delete services.

Figure 7–4 The Sample Services Page



Select an item and...

Select	Name	Description	Visible	Service URL
●	Caltrain	This sample shows how to use location-based modules.	Y	/studio/samples/caltrainNew.jsp
○	HelloWorld	This is a simple Hello, World application.	Y	/studio/samples/helloWorld.jsp
○	OracleCafe	This sample simulates a fast-food ordering system, developed using XHTML/XFORMS.	Y	/studio/samples/OracleCafe.jsp
○	Time	This is a simple application that tells the time.	Y	/studio/samples/time.jsp

7.5.1 Adding a Sample Application

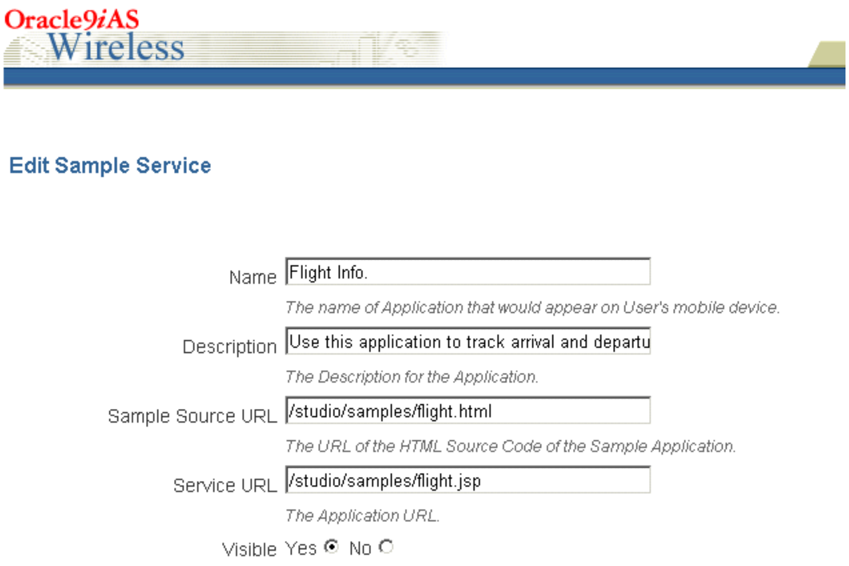
Use the **Add** function to add a service that has already been created and hosted to a location accessible to Mobile Studio.

To add a service, click **Add**. The *Edit Sample Service* page appears (Figure 7–5). Using this page, you specify the name, description, the name of the service JSP and service URL of the new service. Table 7–3 lists the parameters in this page.

Table 7–3 Parameters of the Edit Sample Service Page

Parameter	Value
Name	The name of the service (application) that appears on the end-user’s device.
Description	A description of the service.
Sample Source URL	The HTML document containing the source code for the sample service. The developer of the application should provide this value. This URL must be accessible from Mobile Studio.
Service URL	The service used by the OracleAS Wireless server at runtime for the application. This URL points to the hosting location of the sample service. This URL must be accessible from Mobile Studio.
Visible	The sample service can be hidden from users by setting the *Visible* to *No;* users can view (and use) the service if you set the *Visible* flag to *Yes.*

Figure 7–5 The Edit Sample Services Page (Partial View)



Edit Sample Service

Name	Flight Info.
	The name of Application that would appear on User's mobile device.
Description	Use this application to track arrival and departu
	The Description for the Application.
Sample Source URL	/studio/samples/flight.html
	The URL of the HTML Source Code of the Sample Application.
Service URL	/studio/samples/flight.jsp
	The Application URL.
Visible	Yes ● No ○

Click *Save* to commit your entries and add the service.

7.5.2 Editing a Sample Service

Use the **Edit** button to change the values for a selected service. Clicking the **Edit** button invokes the *Edit Sample Service* page, where you can change the values for the *Name*, *Description*, *Sample Source URL* name, the *Service URL*, and the visibility status of the sample service. Click the *Save* button to store your changes.

7.5.3 Deleting a Sample Service

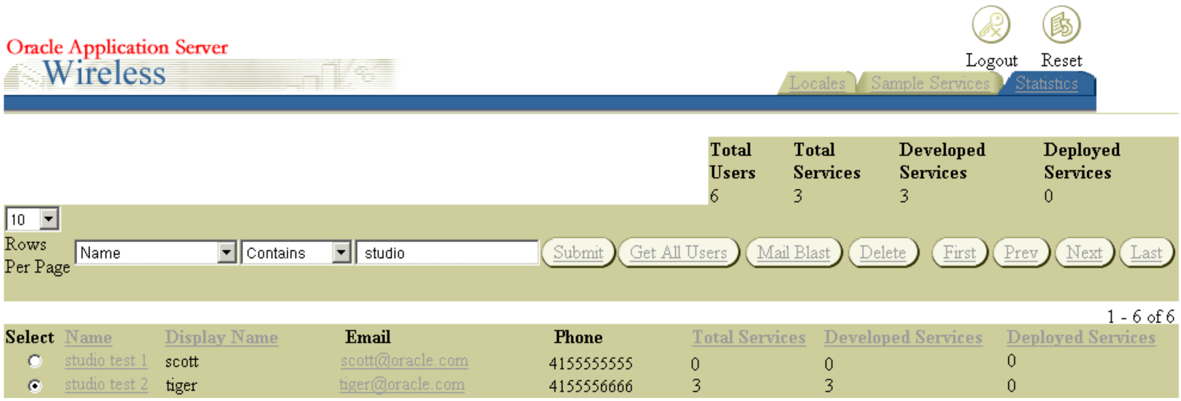
To delete a sample service, first select the sample service from the list of services shown then click the *Delete* button.

> **Note:** You must click *Reset* for any changes made to a sample service to take effect.

7.6 Viewing Statistics

The *Statistics* tab (Figure 7–6) provides user management functions and an overall view of the Mobile Studio in terms of the number of users and services.

Figure 7–6 The Statistics Tab



Total Users	Total Services	Developed Services	Deployed Services
6	3	3	0

10 Rows Per Page

Name | Contains | studio

1 - 6 of 6

Select	Name	Display Name	Email	Phone	Total Services	Developed Services	Deployed Services
○	studio test 1	scott	scott@oracle.com	4155555555	0	0	0
●	studio test 2	tiger	tiger@oracle.com	4155556666	3	3	0

7.6.1 Searching for Users and Services

The *Statistics* tab includes lists that enable you to search for users or services.

To find users, first select the type of search (by name, display, name, company name or e-mail address) and the matching criteria (*is*, *is not*, *starts with*, *ends with*, or *contains*). Next, enter the search string and then click **Submit** to retrieve the search results.

> **Tip:** Clicking **Get All Users** retrieves all Mobile Studio users. Use the *Row Per Page* drop-down to control the length of the search results. Use the **First**, **Prev**, **Next**, and **Last** buttons to navigate through lengthy search results.

Likewise, you query for the number of developed or deployed services for each Mobile Studio user by selecting the Total Services, Developed Services, and Deployed Services options in conjunction with the greater than or less than matching criteria and then by entering a numeric value and clicking Submit.

7.6.2 Notifiying Mobile Studio Users with Mail Blast

You can send a notification to all Mobile Studio users using the **Mail Blast** function. To send a notification, click **Mail Blast** and then enter the message in the page that appears.

> **Note:** The **Mail Blast** function sends a notification to all Mobile Studio users, not just those retrieved by a search.

7.6.3 Deleting Moblie Studio Users

To delete a user, first select the user and then click **Delete**.

8

Managing Foundation Services

This chapter includes the following sections

- Section 8.1, "Overview of the Foundation Management"
- Section 8.2, "Logging In to the Foundation Manager"
- Section 8.3, "Managing Devices"
- Section 8.4, "Managing Transformers"
- Section 8.5, "Managing Adapters"
- Section 8.6, "Managing Regions"
- Section 8.7, "Managing Digital Rights Policies"
- Section 8.8, "Managing API Scan Policies"

8.1 Overview of the Foundation Management

The Foundation Manager enables you to create and modify such objects as devices, transformers, adapters, regions, digital rights policies, and API scan policies in the OracleAS Wireless repository. Table 8–1 describes these objects.

Table 8–1 Objects Created and Managed Using the Foundation Manager

Object Type	Description
Device	A device object associates a physical device or an abstract device with a transformer through user agents and MIME types. A device object captures the device attributes, which are used by both the multi-channel server and the messaging server.
Transformer	A transformer converts the content returned by the OracleAS Wireless adapters. Transformer types include: ■ Result transformers, which convert Adapter Result content into SimpleResult content. ■ Device transformers, which convert SimpleResult content into the final target format. A device transformer can be either the default transformer for a virtual device, or a custom transformer, which is used to render a specific application for a specific physical device.
Adapter	Adapter objects represent the OracleAS Wireless interface to content sources. Adapter objects have an attribute called `classes`, which identify the archive file that contains the actual Java implementation of the adapter.

Table 8–1 (Cont.) Objects Created and Managed Using the Foundation Manager

Object Type	Description
Regions	OracleAS Wireless uses regions to enable developers to assign a location to an application, making the application location-based, unique to a specified area.
Digital Rights Policy	A digital rights policy specifies the execution (or usage) policy of J2ME applications (MIDlets) after users download them. For example, if a downloaded MIDlet can be executed only twice, then you package that application with a digital rights policy to assure that it is executed only twice. Other digital rights policies can be time-based, limiting the execution of MIDlets to prescribed time periods, and be of varying complexity.
API Scan Policy	An API scan policy specifies invalid API calls within J2ME application to the API scan process, which certifies J2ME applications (MIDlets).The invalid APIs are defined with package names, class names and method names in the API scan policy object.

The Foundation Manager provides a set of wizards that enable you to create these objects quickly and with a minimum of information. Each of these wizards break down the creation process into series of steps.

8.2 Logging In to the Foundation Manager

To use the Foundation Manager, you must first access the login page using the following URL:

```
http://<host>:<port>/webtool/login.uix
```

For example, you access the login page by entering the following URL into a browser:

```
http://hostname:7777/webtool/login.uix
```

> **Note:** 7777 is the default port number for Oracle Application Server Wireless.The port number range is 7777 to 7877. To ensure that you are using the correct port number, check the port number for Oracle Application Server Wireless stored in `[Oracle home]/install/portlist.ini`. For more information on port usage, see the Oracle Application Server Administrator's Guide

Enter your user name and password. If you are an administrator, enter *orcladmin* as your user name. (The password is set during installation, but can be changed from the User Manager.)

After you successfully login, select the *Foundation* tab, the Foundation Manager's browsing page appears. From the Foundation Manager, you can administer the following repository objects:

- Devices
- Transformers
- Adapters
- Regions
- Digital Rights Policies
- API Scan Policies

The Foundation Manager provides a tab for each of these repository objects. Each has a browsing page, which enables you to search for an object, as well as access to functions for creating, editing, deleting, and testing.

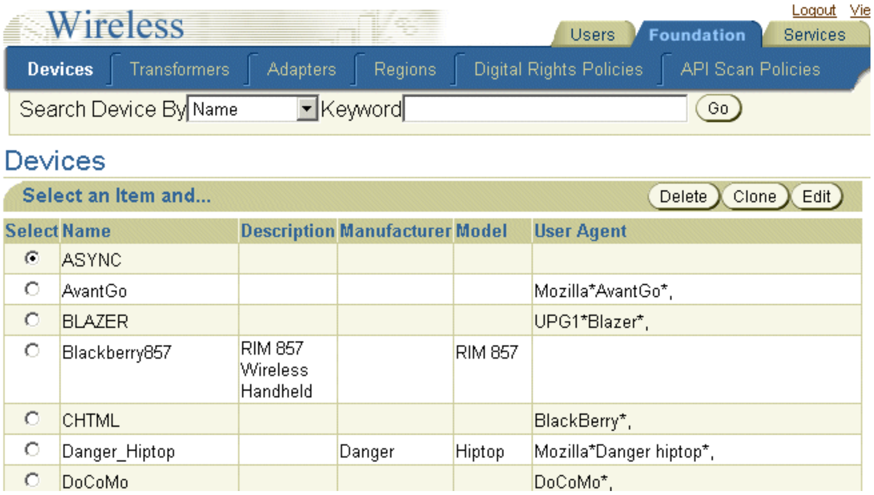
8.3 Managing Devices

A device is an object in the OracleAS Wireless repository that represents either a physical device, such as a Nokia mobile phone, or an abstract device, such as e-mail, which link OracleAS Wireless transformers and the runtime device by recognizing the user-agent, the MIME type, and other HTTP headers.

HTTP headers enable a repository device to map to an actual device. Through the repository device, the OracleAS Wireless Server determines the appropriate transformer for rendering the device result for a variety of browsers, voice gateways, or message clients. For example, if the OracleAS Wireless Server recognizes a device with multi-media display capability, the XMS center (XMSC) renders the multi-media messages bound for the device in images rather than in plain text. Likewise, when delivering a J2ME MIDlet application to a user device, the J2ME provisioning server delivers a version the MIDlet application which is appropriate to the device.

The *Devices* tab enables you to create a device in the repository. Clicking this tab invokes the device browsing page (Figure 8–1), which displays a list of devices in the repository. From this page, you can search for, create, clone, delete, and edit a device.

Figure 8–1 The Browse Devices Page



Search Device By Name Keyword

Devices

Select	Name	Description	Manufacturer	Model	User Agent
●	ASYNC				
○	AvantGo				Mozilla*AvantGo*,
○	BLAZER				UPG1*Blazer*,
○	Blackberry857	RIM 857 Wireless Handheld		RIM 857	
○	CHTML				BlackBerry*,
○	Danger_Hiptop		Danger	Hiptop	Mozilla*Danger hiptop*,
○	DoCoMo				DoCoMo*,

8.3.1 Searching for a Device

From the device browsing page, you can search for devices by keyword, name, manufacturer, model, user agent, or transformer.

To search for a device:

Select one of the following search options:

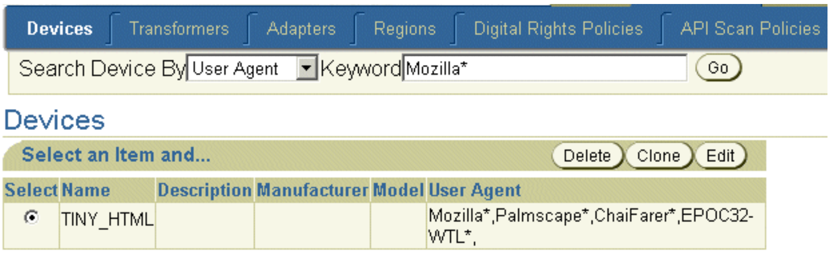
- Name
- Manufacturer
- Model
- Transformer

- User Agent

Enter the keyword for your search.

Click **Go**. The *Search Results* page appears (Figure 8–2).

Figure 8–2 The Search Results Page (for Devices)



Search Device By User Agent Keyword Mozilla*

Devices

Select	Name	Description	Manufacturer	Model	User Agent
●	TINY_HTML				Mozilla*,Palmscape*,ChaiFarer*,EPOC32-WTL*,

8.3.2 Creating a Device

The device creation wizard enables you to create a device by prompting you through each step in the creation process. The wizard dedicates a page to each of these steps; you progress through the wizard by clicking **Next** after completing each step. At any point in the wizard, you can click the **Back** button to return to the preceding pages to change values. You can skip any of the pages in this wizard which contain parameters which do not apply to the device. After you have entered the required information, click **Finish** to complete the device.You can also edit the device to add, remove, or change the parameter values.

To access the wizard, click **Create** in the device browsing page. The wizard appears, defaulting to its first page, where you enter the basic information for the device.

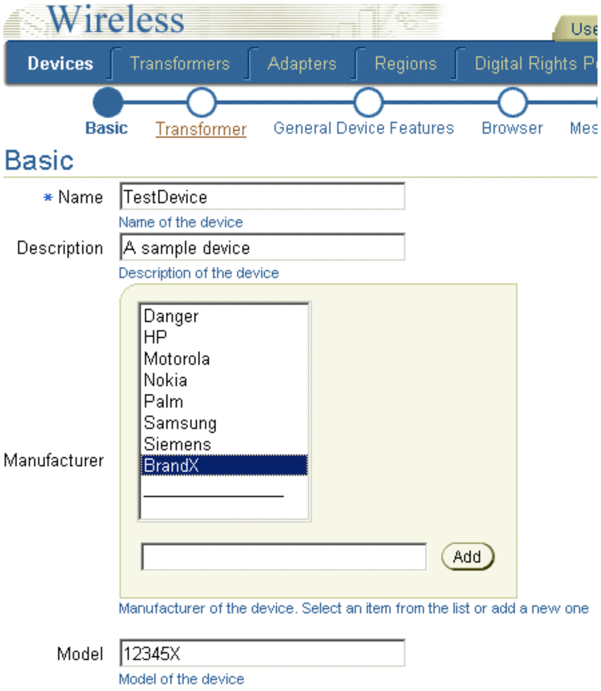
Step 1: Entering the Basic Information for the Device

The *Basic Info.* page (Figure 8–3) enables you to define the general information of the device, such as the device name. Table 8–2 describes the parameters of the *Basic Info.* page.

Table 8–2 Parameters of Basic Information Page

Parameter	Value
Name	The name of the device. This name must be unique. This is a required value.
Description	A description of the device.
Manufacturer	The manufacturer of the device. If the manufacturer does not appear on the list, then enter the name of manufacturer and then click **Add**. The manufacturer then appears on the list, enabling you to select it.
Model	The model number of the device.

Figure 8–3 Entering the Basic Information for a Device



Basic

* Name: TestDevice
Name of the device

Description: A sample device
Description of the device

Manufacturer:
Danger
HP
Motorola
Nokia
Palm
Samsung
Siemens
BrandX
Manufacturer of the device. Select an item from the list or add a new one

Model: 12345X
Model of the device

Step 2: Setting the Transformers

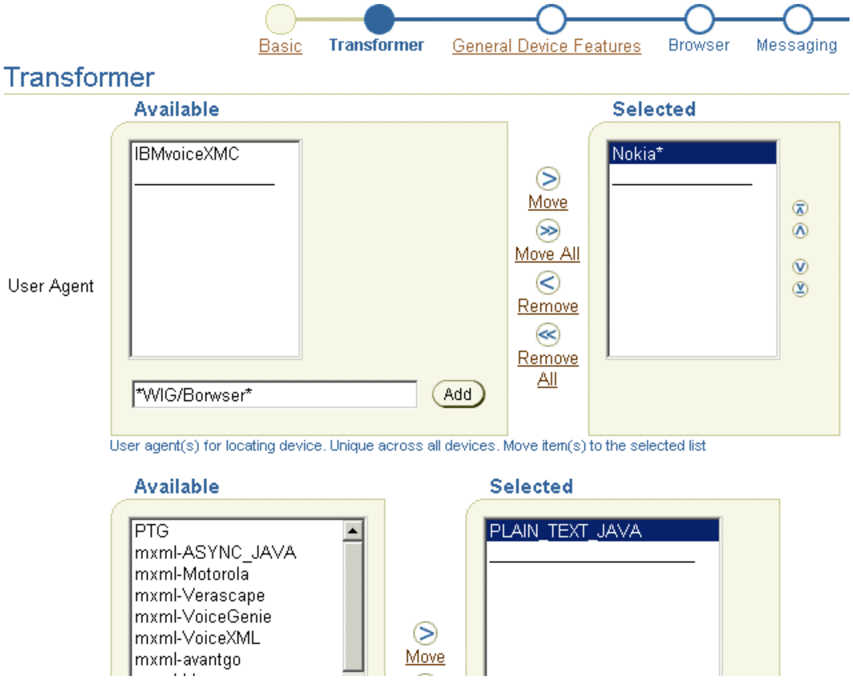
Clicking **Next** in the *Basic Info.* page invokes the *Transformer* page (Figure 8–4). Using this page, you add the transformers and all of the appropriate user agents to the device.

To add user agents, enter the user agents supported by the device and then click *Add*. Continue until you have added all possible user agents for the device. You then select the user agents supported by the device by using the **Move** or **Move All** functions (> and >>) to transfer the user agents from the *Available* pane to the *Selected* pane.

To select transformers for the device, use the **Move** or **Move All** functions (> and >>) to shuttle the transformers from the *Available List* pane to the *Selected List* pane.

Click **Next** after you have selected the user agents and transformers to continue to the next step of the wizard where you device capabilities. Click **Finish** to complete the device.

Figure 8–4 Selecting User Agents and Transformers



Transformer

User Agent

Available: IBMvoiceXMC

Selected: Nokia*

WIG/Borwser

User agent(s) for locating device. Unique across all devices. Move item(s) to the selected list

Available: PTG, mxml-ASYNC_JAVA, mxml-Motorola, mxml-Verascape, mxml-VoiceGenie, mxml-VoiceXML, mxml-avantgo

Selected: PLAIN_TEXT_JAVA

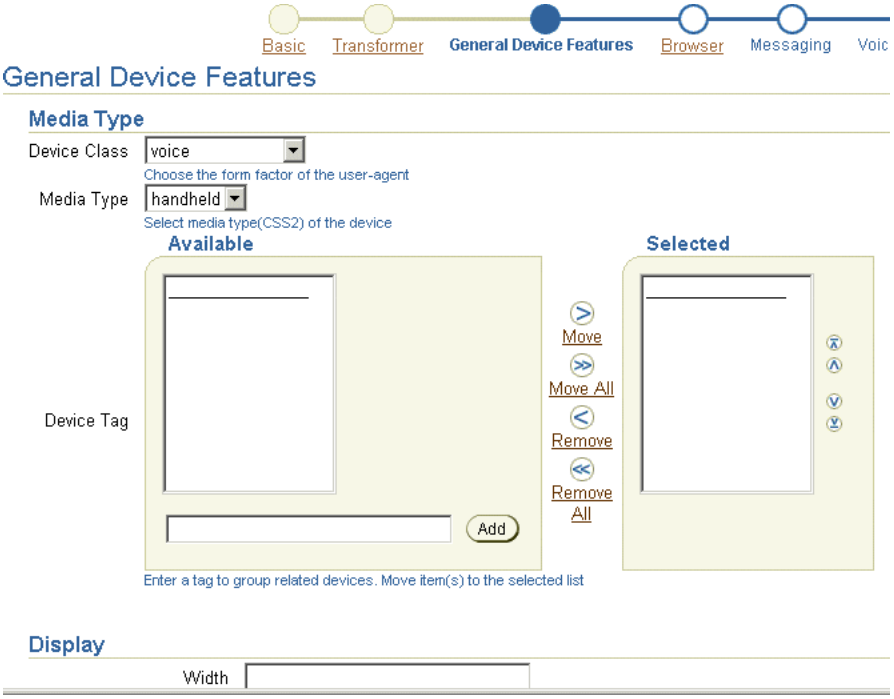
Step 3: Setting Device Capabilities

Device capabilities are categorized into several groups, including general device attributes (media type, display, text input), browser attributes, messaging attributes, voice-grammar attributes, and J2ME attributes (Figure 8–5). OracleAS Wireless examines the values for the device capabilities during the runtime, when the OracleAS Wireless Server renders the device-oriented markup languages, provisions J2ME applications, or sends device-oriented messages. For the detailed explanation of the syntax and semantics of device capabilities, refer to the description of device network adaptation included in the *Oracle Application Server Wireless Developer's Guide*.

Tips:

- Although the device creation wizard provides separate pages for the device capabilities, none of these related parameters are required; you can successfully complete a device if you do not define any of these parameters.
- To help you enter the values for the device capabilities parameters, the *Device Capabilities* pages include in-line help as hints under each of the inputting fields.You can also refer to the online help. On any of the *Device Capabilities* pages, you can click **Finish** to complete the device and skip the remaining steps.

Figure 8–5 Entering Device Capabilities -- Entering the General Device Features



General Device Features

Media Type

Device Class: voice

Choose the form factor of the user-agent

Media Type: handheld

Select media type(CSS2) of the device

Enter a tag to group related devices. Move item(s) to the selected list

Display

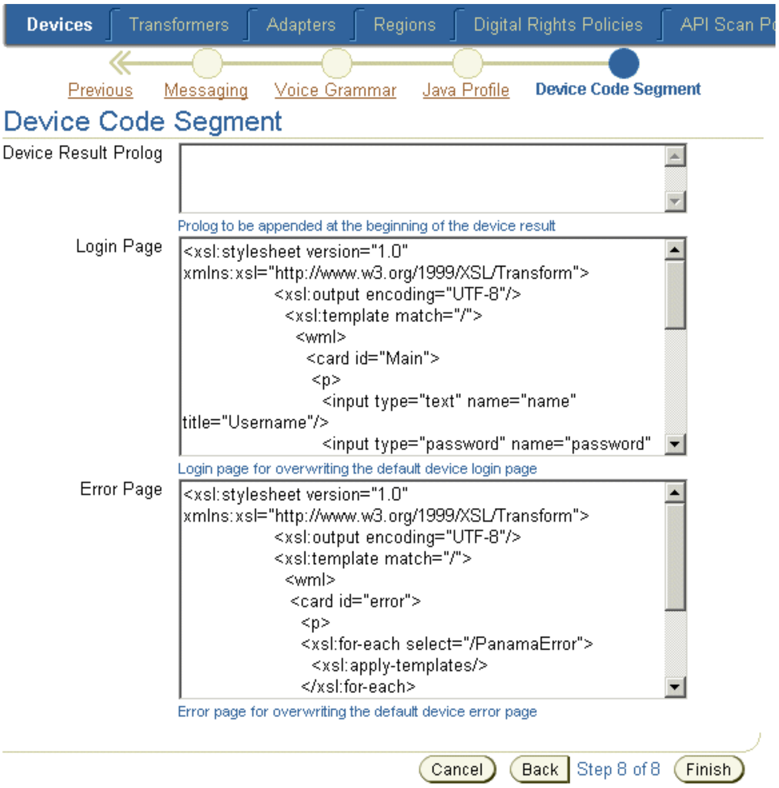
Width

Step 4: Setting the Device Code Segment

Using the *Device Code Segment* page (Figure 8–6), you enter the device result prolog, the login page, and the error page.

For the device result prolog, you enter the code segment which is added to all the rendering results for this device. Entering a login page replaces the device’s default login page. Likewise, entering an error page replaces the default error page for the device. Click **Finish** to complete the device

Figure 8–6 Entering the Device Result Prolog, Login and Error Pages



Device Code Segment

Device Result Prolog

Prolog to be appended at the beginning of the device result

Login Page

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
                <xsl:output encoding="UTF-8"/>
                  <xsl:template match="/">
                    <wml>
                      <card id="Main">
                       <p>
                         <input type="text" name="name"
title="Username"/>
                         <input type="password" name="password"
```

Login page for overwriting the default device login page

Error Page

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
                <xsl:output encoding="UTF-8"/>
                <xsl:template match="/">
                  <wml>
                   <card id="error">
                     <p>
                     <xsl:for-each select="/PanamaError">
                       <xsl:apply-templates/>
                     </xsl:for-each>
```

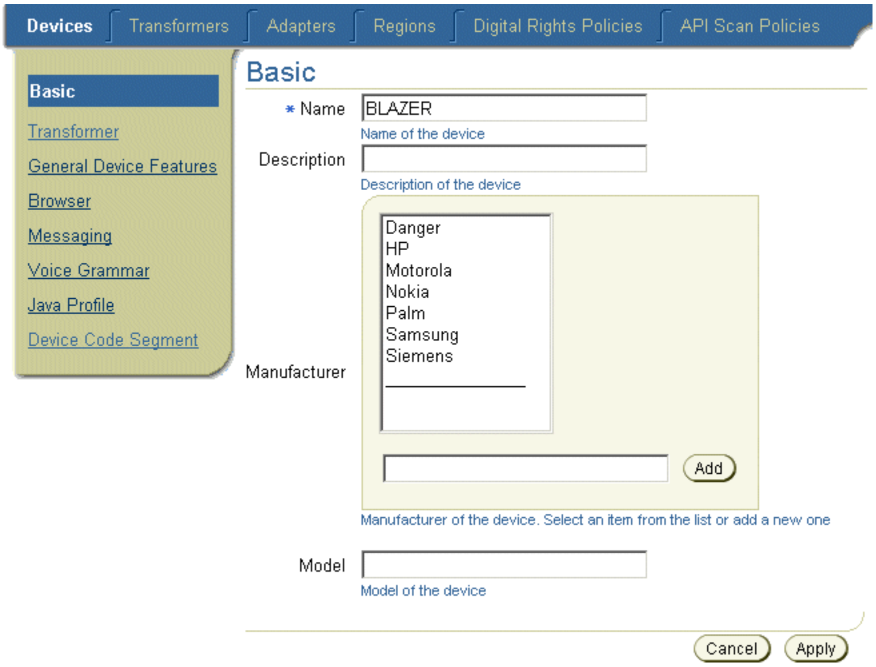
Error page for overwriting the default device error page

8.3.2.1 Editing a Device

The **Edit** button in the device browsing page enables you to edit all of the information of a device. To edit a device, first select the device and then click the **Edit** button. The editing page appears and defaults to the parameters defined for the basic information of the device (Figure 8–7). You can select other device properties by selecting the appropriate links in the menu on the left side of the editing page. Click **Apply** to save any changes that you make to the parameters. Clicking **Cancel** returns you to the device browsing page.

Refer to the steps described in Section 8.3.2 for descriptions of the parameters for creating a device.

Figure 8–7 Editing a Device



Basic

* Name: BLAZER

Name of the device

Description:

Description of the device

Manufacturer: Danger, HP, Motorola, Nokia, Palm, Samsung, Siemens

Manufacturer of the device. Select an item from the list or add a new one

Model:

Model of the device

8.3.2.2 Deleting a Device

You delete devices from the repository by selecting a device from the browsing page and then clicking **Delete**.

8.3.3 Cloning a Device

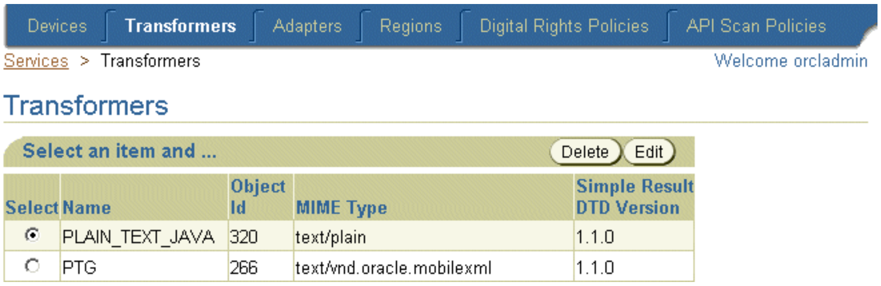
The **Clone** function enables you to create a new a new device with properties similar to an existing device; the new device inherits all of the capabilities from the existing device from which it was copied. Unlike creating a new device as described in Section 8.3.2, you need only enter a name for the device.You can later edit the parameters for the cloned device.

To clone a device, you select a device from the browsing page and then click **Clone**. Enter a name for the new device and then click **Finish**.

8.4 Managing Transformers

Clicking the *Transformers* tab displays the browsing page for transformers, which includes a table that lists the current transformers in the repository by name, object ID in the repository, the MIME type supported by the transformer, and the **Simple Result format** DTD version. Figure 8–8 illustrates this browsing page.

Figure 8–8 The Browse Transformers Page (Partial View)



Transformers

Select	Name	Object Id	MIME Type	Simple Result DTD Version
●	PLAIN_TEXT_JAVA	320	text/plain	1.1.0
○	PTG	266	text/vnd.oracle.mobilexml	1.1.0

From this page you can delete, edit, and create transformers.

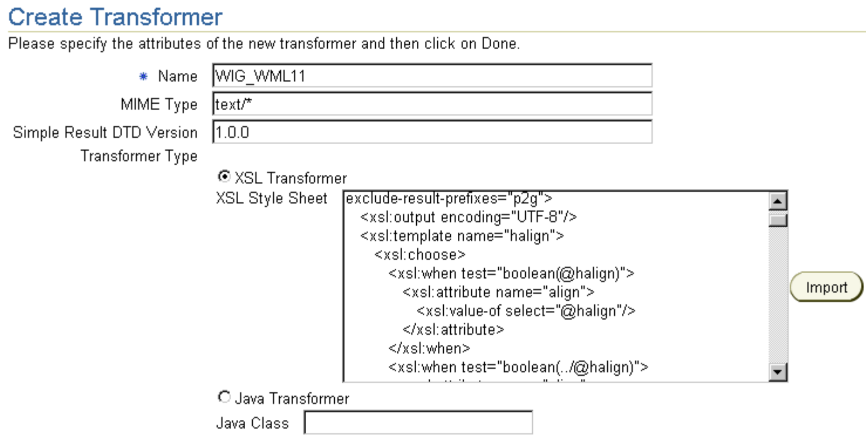
8.4.1 Creating a Transformer

To create a transformer, click the **Create Transformer** button to invoke the *Create Transformer* page (Figure 8–9). To complete the transformer, you must define the following parameters, described in Table 8–3 and then click **Finish**.

Table 8–3 Parameters of the Create Transformer Page

Parameter	Value
Name	The name of the transformer. This name must be unique.
MIME Type	The MIME type that the transformer supports.
SimpleResult DTD Version	The SimpleResult DTD version, such as 1.0.0 (the default version).
Java Transformer	Specifies a Java class transformer implementation.
Class Name	The name of the class that implements the transformer.
XSL Transformer	Specifies an XSLT style sheet transformer implementation. If you select an XSL transformer, you can do one of the following: ■ Enter the code for the XSL style sheet in the field next to the *Style sheet* parameter, then click **Finish**. ■ Using a text editor, open an existing XSL style sheet, copy and paste the lines that you want to use, and then click **Finish**. ■ Click the **Import** button to import an existing XSL style sheet.
XSL Stylesheet	The actual XSLT style sheet that implements the transformer. You can cut and paste a transformer from another editing environment into this field.
Java Transformer	Specifies a Java class transformer implementation.
Java Class	The name of the class that implements the transformer.

Figure 8–9 The Create Transformer Page



Create Transformer

Please specify the attributes of the new transformer and then click on Done.

* Name	WIG_WML11
MIME Type	text/*
Simple Result DTD Version	1.0.0
Transformer Type	

● XSL Transformer

XSL Style Sheet

```
exclude-result-prefixes="p2g">
  <xsl:output encoding="UTF-8"/>
  <xsl:template name="halign">
    <xsl:choose>
      <xsl:when test="boolean(@halign)">
        <xsl:attribute name="align">
          <xsl:value-of select="@halign"/>
        </xsl:attribute>
      </xsl:when>
      <xsl:when test="boolean(../@halign)">
```

○ Java Transformer

Java Class

8.4.2 Editing a Transformer

To edit a transformer, select a transformer from the browsing page and then click **Edit.** The editing page appears, with its fields populated with the values defined for the selected transformer. Clicking **Apply** saves any changes. Clicking **Cancel** sets the parameters back to their previous values and returns you to the browsing page.

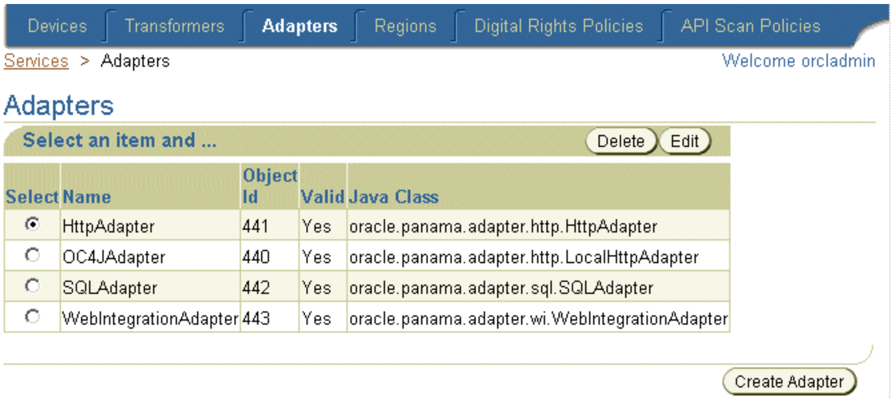
8.4.3 Deleting a Transformer

To delete a transformer, select a transformer from the browsing page and then click **Delete**.

8.5 Managing Adapters

Selecting the *Adapters* tab invokes the browsing page for the adapters (Figure 8–10). This page includes a table which lists the current adapters by their object IDs in the repository, their status as valid adapters (that is, adapaters which are available to master applications) and by the Java class that either implements the adapter, or serves as an entry point to the classes that implement the adapter.

Figure 8–10 Partial View of the Browse Adapters Page



Adapters

Select	Name	Object Id	Valid	Java Class
●	HttpAdapter	441	Yes	oracle.panama.adapter.http.HttpAdapter
○	OC4JAdapter	440	Yes	oracle.panama.adapter.http.LocalHttpAdapter
○	SQLAdapter	442	Yes	oracle.panama.adapter.sql.SQLAdapter
○	WebIntegrationAdapter	443	Yes	oracle.panama.adapter.wi.WebIntegrationAdapter

You use this page to create, edit, and delete adapters.

8.5.1 Creating an Adapter

To create an adapter, click *Create Adapter* in the browsing page. The *Create Adapter* page appears. To create an adapter, you must define the following parameters, which are described in Table 8–4.

Table 8–4 Parameters of the Create Adapter Page

Parameter	Value
Name	The name of the adapter. The name must be unique.
Valid	Specifies whether the adapter is available to the master applications. If selected, the adapter is available. If this option is clear, then the adapter is invalid and therefore unavailable. As a result, all of the master applications that use the adapter are also invalid.
Java Class	The Java class that either implements the adapter or serves as the entry point for the classes that implement the adapter.

After you enter the needed parameters, click **Create**. The browsing page reappears, displaying the new adapter. Clicking **Cancel** clears any values entered and returns you to the browsing page.

8.5.2 Editing an Adapter

To edit an adapter, select the adapter from the browsing page and then click **Edit**. The *Edit Adapters* page appears, displaying the values for the selected adapter. Click **Apply** to commit your changes. Clicking **Cancel** clears any values entered and returns you to the browsing page.

8.5.3 Deleting an Adapter

To delete an adapter from the repository, select the adapter and then click **Delete**.

8.5.4 Setting Adapter Parameters

The following sections describe the uses and parameters of the OracleAS Wireless adapters.

- Section 8.5.4.1, "Setting the Initialization (Init) Parameters for Adapters"
- Section 8.5.4.2, "Setting the Input Parameters for Adapters"

8.5.4.1 Setting the Initialization (Init) Parameters for Adapters

When you create a non-HTTP master application, the *Init Parameters* page of the *Master Application Creation Wizard* shows the initialization (init) parameters specific to the type of adapter selected for the master application. When OracleAS Wireless first invokes the adapter, it passes the values that you set in the Init Parameters page to the adapter.

8.5.4.1.1 Setting Init Parameters for the SQL Adapter The SQL adapter retrieves and adapts content from any JDBC-enabled data source for a master application based on the SQL adapter, the *Init Parameters* panel includes the following parameters, which are described in Table 8–5.

Table 8–5 Init Parameters for the SQL Adapter

Parameter	Value
JDBC Connect String	The JDBC connect string for the database on which to query, as follows: `jdbc:oracle:thin:@host_name:port:SID` **Note**: Insert all colons (for example, *thin:@host*).
JDBC Driver	The Java DriverClass name (for example, Oracle thin driver, `Oracle.jdbc.driver.oracle.driver`)
User Name	The name of the database user.
Password	The password of the database user
Type of Statement	The type of SQL statement used by the master application. Allowable values include: QUERY: for a select statement. This type of statement returns a Simple Result document. You can use output filtering with QUERY statements. PLSQL: to use a PL/SQL procedure. This type of statement returns results to a database buffer. CALL: to run a stored procedure (SQL92 syntax only). This returns either a Simple Result or an Adapter Result element.
The Statement	The actual SQL statement that invokes the query, PL/SQL procedure, or stored procedure. **Note**: The SQL statement should be entered without a semicolon. You can use input variables in the SQL statement. You must indicate a variable in the statement by prefixing the variable with a colon. For example, you can specify an input variable in a PL/SQL statement as follows: `begin mypackage.foo(:expr); end;` Where `:expr` is the name of the variable. You must define the parameter manually in the input panel.
Minimum DB Connection Pool Size	The minimum number of database connections.
Maximum DB Connection Pool Size	The maximum number of database connections.
Increment Size for the Connection Pool	The increment by which the database connection pool increases.
Idle Timeout (In Minutes)	The time (in minutes) of inactivity that OracleAS Wireless allows before automatically logging the user off the system.

8.5.4.1.2 Setting Init Parameters for the Web Integration Adapter The Web Integration adapter retrieves and adapts Web content. The Web Integration adapter works with Web Interface Definition Language (WIDL) files to map source content to OracleAS Wireless XML. Typically, the source format for the Web Integration adapter is HTML, but developers can also use the adapter to retrieve content in other formats, such as XML.

Table 8–6 describes the initialization (init) parameters for a master application based on the Web Integration adapter.

Table 8–6 Init Parameters of the Web Integration Adapter

Parameter	Value
WebIntegrationServer	The machine name and listening port of the Web Integration Server. If the Web Integration Server and the OracleAS Wireless server reside on the same machine, use `localhost:port`. This field is required. The server you specify in this field must be running for the Content Manager to return the adapter parameters.
Interface	The WIDL interface name. This interface must be published to the Web Integration Server. You can publish the interface using the Web Integration Developer. You cannot currently use the WIDL_FILE parameter to identify a WIDL service.
WIDL_FILE	Do not enter a value for this parameter.

8.5.4.1.3 Setting Input Parameters for the Web Integration Adapter The master application determines the parameters that display in the panel by querying the adapter. Every input parameter defined in the WIDL interface appears in the Inputs panel, including parameters for other WIDL applications within the WIDL interface.

In addition to the custom input parameters that you create, Web Integration applications provide these parameters:

- `OutputType`
- `PAsection`
- `InputEncoding`

The `OutputType` specifies the type of XML output that the adapter should return. You can specify `RawResult`, to return content in **Adapter Result format**, or `SimpleResult`, to return content in **Simple Result format**. For returning the raw result format, you must create a result transformer that converts the result into Simple Result for the device transformer. The result transformer should have the same name as the value you use for the `PAsection` parameter; that is, it should have the same name as the WIDL application.You can use `RawResult` for chained services.

`PAsection` is the name of the WIDL application that you want the master application to invoke. A WIDL interface can include more than one WIDL application. OracleAS Wireless lists the WIDL application names in a selection list in the value field.

`InputEncoding` specifies the encoding used to encode the source document. The source document is the URL that was used to create the WIDL file for this application. The default value of this parameter is UTF-8. If the language of the source document is an Asian language, you can change the default encoding to the appropriate multi-byte encoding according to the IANA standards for the particular Asian language that is used in the source document. The `InputEncoding` parameter enables you to specify or change the encoding as part of the multi-byte character support.

8.5.4.2 Setting the Input Parameters for Adapters

The *Input Parameters* page displays the input parameters for the adapter. The OracleAS Wireless Server queries the adapter definition to determine the parameters that appear in this panel. The master application passes the input parameter values to the adapter’s invoke method every time the adapter executes.

Some parameters rely on user input for values. The values for other parameters, such as name of the WIDL application in the WIDL interface (`PAsection`), are set by the master application or application link. `PAsection` is an internal parameter, not exposed to the end user. In addition to `PAsection`, OracleAS Wireless provides these input parameters, which are described in Table 8–7.

Table 8–7 Input Parameters for a Non-Http Master Application

Parameter	Value
`PAservicepath`	The relative path to a OracleAS Wireless application, such as `/UsersFolders/joe/myChain`.
`PAdebug`	The debugging option. If set to *true* (that is, the value is set to *1*), then OracleAS Wireless produces verbose output to the log files. For verbose logging, OracleAS Wireless writes the results of adapter invocations to the log file along with notifications and warnings. This enables you to examine application content in its internal, XML format, which can help you to create result transformers and solve application and transformer problems.
`PAsection`	The WIDL adapter uses this value to identify the application that serves as the entry point in the chained application sequence.
`PAuserid`	The user name.
`PApassword`	The user password.
`PAsid`	The OracleAS Wireless session identifier.

Table 8–8 describes the OracleAS Wireless input parameters.

Table 8–8 Input Parameter Attributes

Parameter	Value
Name	The name of the input parameter. The OracleAS Wireless sets the name of the input parameter by querying the adapter definition.
Caption	The caption is the label that OracleAS Wireless uses for the parameter when prompting for user input.
Comment	In the case of master applications based on the Web Integration adapter, OracleAS Wireless automatically populates this field with the name of the WIDL application that uses the parameter. For applications based on other adapters, you can use this column to document the parameter. The comment is only used internally.
User Customizable	Specifies whether the end user can set a value for this parameter using OracleAS Wireless Customization. You can make most input parameters customizable by the user. In particular, you should set this option for parameters that may be difficult for a user to enter from a mobile device. This includes e-mail addresses and personal identification numbers.

Table 8–8 (Cont.) Input Parameter Attributes

Parameter	Value
Format	This mask sets the expected data entry mode for the user device. For example, if you expect the user to enter numbers for the parameter, you use the format code *N*. This works only with WML 1.1-compliant devices. The default format is *M (all formats). Other formats include: ■ A, for entry of uppercase letters or punctuation ■ a, for entry of lowercase letters or punctuation ■ N, for entry of numeric characters. ■ X, for entry of uppercase letters. ■ x, for entry of lowercase letters. For a complete list of formats, see the *Wireless Application Protocol Wireless Markup Language Specification, Version 1.1*.
Mandatory	Select this check box if this parameter must have a value. Remove the selection for optional parameters.
Default Value	For most parameters, this value represents the default value for the parameter. If you specify a default value, OracleAS Wireless does not prompt the user for a value. Default values can be overridden by a value specified by an application link created by the Content Manager or, if the parameter is visible to the user in the OracleAS Wireless Customization Portal. The `PAsection` parameter is used by the Web Integration adapter. For `PAsection`, this value is the name of the WIDL application that the Web application should use. You can select the names from a selection list. If you do not specify a value for `PAsection`, the OracleAS Wireless application includes all WIDL applications in the WIDL interface.

8.5.4.3 Adding a New Input Parameter to the Adapter

From the *Input Parameter* page click **Add Another Row**. A blank row appears. Define the name for the input parameter and any other needed parameters in this row, which are described in Table 8–8.

8.5.4.3.1 Setting Input Parameters for the AppsFramework Adapter The AppsFramework adapter enables the development of enterprise applications on top of Wireless. It provides system-wide standard application look and feel, enhanced application widgets support and data binding to enterprise data.

The AppsFramework adapter includes the input parameter classname which must be the package and class of the implementation of the `MobileApplicationHandler` interface.

8.5.4.3.2 Modifying the Style, Color, and SDU Information for the Mobile Application Framework Adapter The Mobile Application Framework adapter uses style and color mappings to provide a uniform look and feel that can be customized across all applications running on the server. In addition, carrier-specific information can be specified to the Mobile Application Framework adapter to optimize the content delivered by the adapter. The StyleColorLoader command-line utility is used to modify the style, color, and SDU size information used by the Mobile Applications Framework adapter.

Downloading the Style/Color/SDU Repository

To download the Style/Color/SDU Repository:

1. Change directory to `ORACLE_HOME/wireless/sample`

2. Enter `updateStyleColor.bat -D <filename>`, where `<filename>` is the target file that receives the downloaded XML repository. For a UNIX system, enter `updateStyleColor.sh -D <filename>`.

Uploading the Style/Color/SDU Repository

To upload the Style/Color/SDU/Repository:

1. Change directory to ${ORACLE_HOME}/wireless/sample.
2. Enter `updateStyleColor.bat -U <filename>`, where `<filename>` is the file containing the Style/Color/SDU information in the specified XML format that should be uploaded into the database. On a UNIX system, enter `updateStyleColor.sh -U <filename>`.

Modifying the Style/Color/SDU XML Repository File

To modify the Style/Color/SDU XML repository file:

1. Download the file.
2. Modify this file by opening it in any text editor. The XML file contains three top-level elements: `<StyleSet>`, `<ColorSet>`, `<SDUSize>`. After making modifications, you then upload the file back into the repository.

Defining a StyleSet

The `<StyleSet>` elements help the renderers for a given device render application styles into markup language, as described above. For example, if you want to create a prompt- style "Prompt" and bind the style to the text of the prompt, you create a "Prompt" style in the style repository.

Each `<StyleSet>` element contains a number of `<Style>` elements. Each `<Style>` element contains a name, a font face, font size, font style, and font color. Table 8–9 describes the style element properties.

Table 8–9 Style Element Properties

Property Name	Required?	Multiple?	Description
Name	Yes	No	The name of the Style.
FontFace	Yes	No	The name of the font face of the given style.
FontSize	Yes	No	The font size of the given style.
FontColor	Yes	No	The name of the font color of the given style.
FontStyle	Yes	No	The name of the font style of the given style, (that is, *Bold*, *Italic*, *Plain*).

In addition to the `<Style>` element, the StyleSet contains elements described in Table 8–10.

Table 8–10 StyleSet Element Properties

Property Name	Required?	Multiple?	Description
Name	Yes	No	The name of the `<StyleSet>`. If a StyleSet is not associated with the device, then the `<StyleSet>` named `Default` is assigned to the device.
Inherits	Yes	No	The parent style sheet from which style definition are inherited.Often, the administrator wants only to change a single style between two devices. In such a case, the administrator defines a single `<StyleSet>`, which has all of the style definitions for the first device. The second device then inherits this `<StyleSet>` and only overwrites the styles that are different between the two `<StyleSet>` elements.
Style	Yes	No	This element defines a style.
Device	Yes	No	Describes the type of devices associated with a style set. The two types of devices supported are Phone and PDA.

By modifying application style definitions in a given `<StyleSet>`, the system administrator can control how the given application style is rendered on the device to which the style set is bound across the whole system. For example, if a PDA device is bound to the StyleSet *Default*, then changing the prompt style in the default StyleSet to bold from plain results in all prompts appearing in bold rather than in plain when rendered on client devices in the PDA device grouping.

Defining a ColorSet

The `<ColorSet>` element helps the renderers for a device render application colors into markup language. For a given device, this application color is mapped to a color code, which can be modified by the system administrator to produce the optimal rendering. For example, if a PDA device is bound to the `<ColorSet>`, *Default*, then changing the background color in the default `<ColorSet>` to grey from white results in the background color for all applications on client devices in the PDA device grouping to be grey rather than white.

A `<ColorSet>` element consists of multiple `<Color>` elements. The following table describes the propertis common to each `<ColorSet>`.

Table 8–11 ColorSet Elements Properties

Property Name	Required?	Multiple?	Description
Name	Yes	No	The name of the `<ColorSet>`.
Inherits	Yes	No	The parent `<ColorSet>` from which color traits are inherited. Often, an administrator wants only to change a single application color between two devices. In this case, the administrator defines a single color set which has all of the color definitions for the first device.This color set is then inherited by the second device, which would only overwrite the colors that are different between the two `ColorSet` elements.
Color	Yes	Yes	This element defines a color.
Device	Yes	No	Describes the type of device associated with the style set. The two types supported devices are PDA and Phone.

A `<ColorSet>` element consists of multiple `<Color>` elements. The following table describes the properties common to all `<Color>` elements.

Table 8–12 ColorSet Color Element Properties

Property Name	Required?	Multiple?	Description
Name	Y	N	The name of the Style.
ColorDesc	Y	N	The 24-bit color code of the given color, for example White = #FFFFFF.

Defining SDUSize Information for a Device

The `<SDUSize>` element enables the renderers for a given device to render an optimized amount of information on pages. For a given device, the `<SDUSize>` is the upper limit on the amount of information (in bytes) that the network can carry to this device.

A `<SDUSize>` element consists of two child elements. The following table lists their properties.

Table 8–13 SDUSize Element Properties

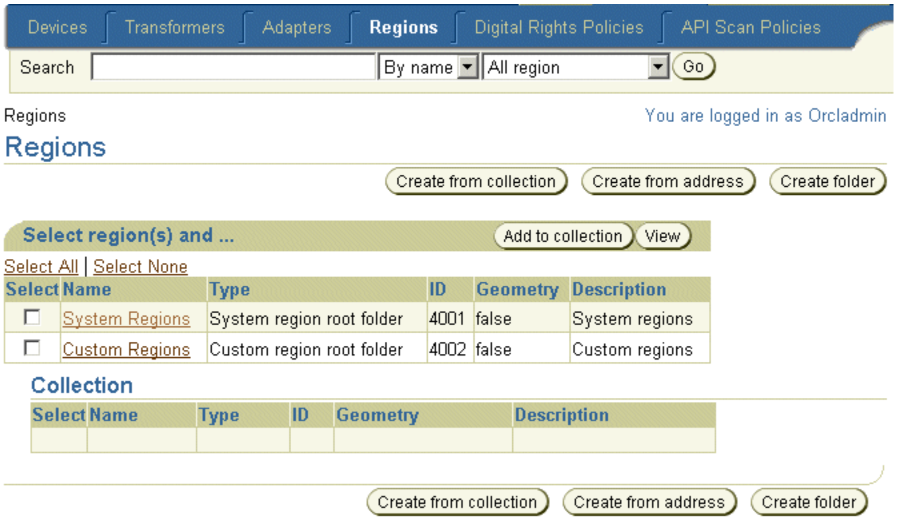
Property Name	Required?	Multiple?	Description
Name	Yes	No	The name of the type of device. The two types of devices supported are Phone and PDA.
Value	Yes	No	The 24-bit color code of the given color, for example White = #FFFFFF.

8.5.4.3.3 Setting Input Parameters for the SQL Adapter You can configure SQL input parameters in the same way that you configure the Web service parameters. You specify input parameters in the SQL statement you use to implement the service.

8.6 Managing Regions

When you click the Regions tab in the Foundation Manager, the main display of the region modeling tool appears (Figure 8–11).

Figure 8–11 The Main Display of the Region Modeling Tool



Regions

You are logged in as Orcladmin

Regions

Select region(s) and ...

Select	Name	Type	ID	Geometry	Description
□	System Regions	System region root folder	4001	false	System regions
□	Custom Regions	Custom region root folder	4002	false	Custom regions

Collection

Select	Name	Type	ID	Geometry	Description

The region modeling tool enables administrators of a wireless portals to create custom regions that can be associated with location-based applications.

You create a location dependent application by specifying a region. This region can be a system-defined region (one provided out-of-the-box with OracleAS Wireless) or a custom region, one created with the region modeling tool.

A region is a geographic entity, or location. A region can be small (such as a street address) or large (such as a country). A region can be represented by a point, as is often done for addresses and locations of interest (such as airports and museums), or by a polygon, as is usually done for states and countries. For detailed information about using the region modeling tool, refer to the chapter on location services in the Oracle Application Server Wireless Developer's Guide.

8.7 Managing Digital Rights Policies

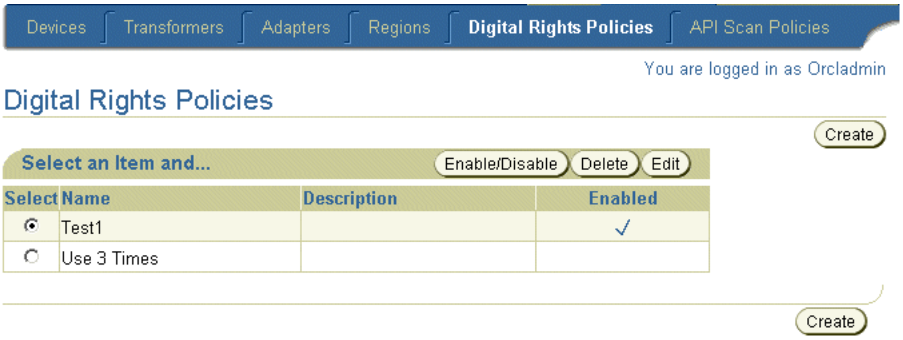
A digital rights policy restricts the execution of J2ME applications on mobile devices. Out of the box, OracleAS Wireless provides two types of digital rights management (DRM) policies that can be used to package J2ME applications: *Count DRM policy*, and *Interval DRM* policy.

The *Count DRM* policy restricts the number of times that a downloaded J2ME application can be run on a device. The *Interval DRM* policy sets the period in which a downloaded J2ME application can be run on a device from the time the user downloads the application. In addition, OracleAS Wireless provides a platform to create a customized digital rights policy.

All digital rights policies created using the Foundation Manager can be selected from the Content Manager when creating an application link based on a J2ME application. For more information, see Section 6.3.4.

Use the Digital Rights Policy subtab to manage the digital rights policies. When you click the Digital Rights Policy subtab, the browsing page for digital rights policies appears (Figure 8–12), displaying a list policies in the repository.

Figure 8–12 The Browsing Page for Digital Rights Policies



Select	Name	Description	Enabled
●	Test1		✓
○	Use 3 Times		

8.7.1 Creating a Digital Rights Policy

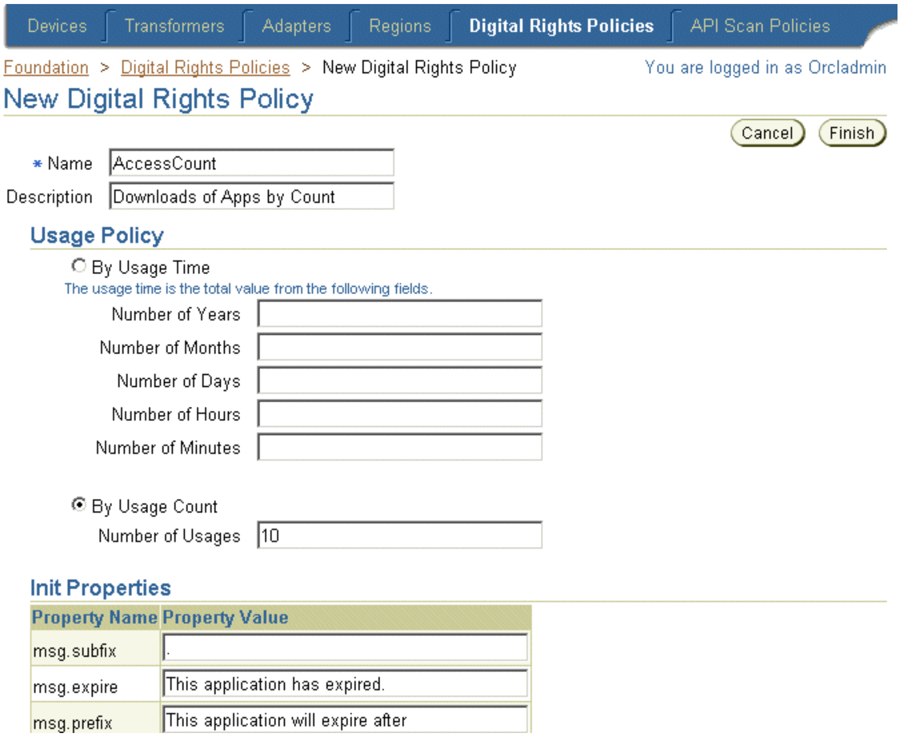
OracleAS Wireless provides a two-step wizard which enables you to create a digital rights policy. To access this wizard, click **Create** in the browsing page.

Step 1: Selecting the Digital Rights Policy Package Type

There are two types of digital rights policy packages: one is a default package provided by OracleAS Wireless; the other is a customized package that you can plug into the OracleAS Wireless platform. If you select this customized package, then you must specify the full class name of the packaging class, which implements the `oracle.wireless.me.server.tools.drm.DRMPackager` interface.

To create a digital rights policy, click **Create**. The page for the policy's attributes page appears (Figure 8–13).

Figure 8–13 Entering the Attributes for a Digital Rights Policy


Devices
Transformers
Adapters
Regions
Digital Rights Policies
API Scan Policies
Foundation > Digital Rights Policies > New Digital Rights Policy
You are logged in as Orcladmin
Cancel
Finish

New Digital Rights Policy

* Name: AccessCount

Description: Downloads of Apps by Count

Usage Policy

○ By Usage Time

The usage time is the total value from the following fields.

Number of Years	
Number of Months	
Number of Days	
Number of Hours	
Number of Minutes	

● By Usage Count

Number of Usages: 10

Init Properties

Property Name	Property Value
msg.subfix	.
msg.expire	This application has expired.
msg.prefix	This application will expire after

Step 2: Entering the Digital Rights Policy Detail Attributes

If you selected the **Default Package**, then you must specify the following attributes:

- A name for the digital rights policy. This is a required parameter.
- A description for this digital rights policy. This is an optional parameter.

Selecting the Usage Policy

To limit the number of times that a user can execute a downloaded J2ME application, define the values for the **Usage Time**, or **Usage Count** options.

For the **Usage Time** option, specify the number of years, months, days, hours or minutes that the user can execute the downloaded application. Define the **Usage Count** option by specifying the number of times that a user can execute a downloaded J2ME application.

Entering the Initialization Properties

Each time that the user executes an application, a message displays on the user's device informing the user of the remaining number of times (or the remaining amount of time), that the user has to access the application. To create such a message, define the *msg.subfix*, *msg.expire*, and *msg.prefix* parameters. Table 8–14 describes these parameters, which enclose the usage count display presented to the user for each download.

Table 8–14 Initialization Parameters of a Digital Rights Policy

Parameter	Value
msg.subfix	The punctuation and text that follow the usage count data. For example, enter *times.*
msg.expire	The text telling the user that the application has expired, or is no longer available. For example, enter *This application has expired*!
msg.prefix field	The text that precedes the user count display. For example, enter *This application expires after* [times].

Click **Create** to complete the policy.

Defining a Customized Package

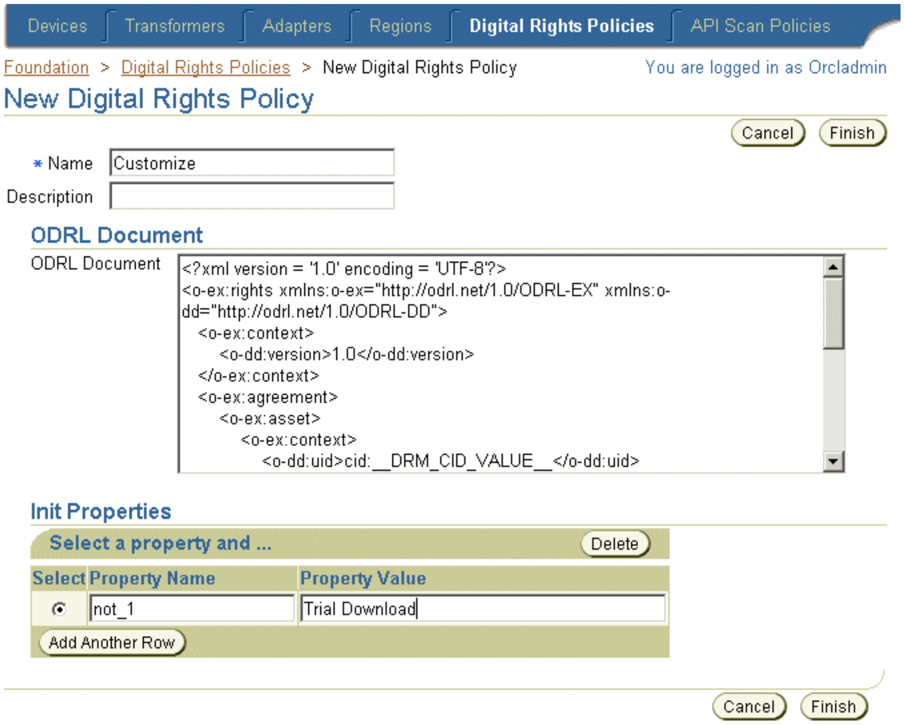
If you selected the **Customized Package** option in Step 2, then you must define a name for the digital rights policy and optionally enter a description for the policy in the New Digital Rights Policy page (Figure 8–14).

You can also enter an Open Digital Rights Language (ORDL) document, an XML document which expresses the Digital Rights Policy. This ODRL document is consumed by the packaging object which implements `oracle.wireless.me.server.tools.drm.DRMPackager`.

In addition, you can enter the initialization (init) properties associated with the policy. The init property name and value pairs are passed to Custom Digital Right implementation class. This implementation class uses these value pairs.

Click **Finish** to complete the policy.

Figure 8–14 Defining a Customized Package


Devices
Transformers
Adapters
Regions
Digital Rights Policies
API Scan Policies
Foundation > Digital Rights Policies > New Digital Rights Policy
You are logged in as Orcladmin
Cancel
Finish

New Digital Rights Policy

* Name: Customize

Description:

ODRL Document

ODRL Document

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.0/ODRL-EX" xmlns:o-
dd="http://odrl.net/1.0/ODRL-DD">
  <o-ex:context>
    <o-dd:version>1.0</o-dd:version>
  </o-ex:context>
  <o-ex:agreement>
    <o-ex:asset>
      <o-ex:context>
        <o-dd:uid>cid:__DRM_CID_VALUE__</o-dd:uid>
```

Init Properties

Select	Property Name	Property Value
●	not_1	Trial Download

Select a property and ...
Delete
Add Another Row
Cancel
Finish

8.7.2 Editing a Digital Rights Policy

The **Edit** button in the digital rights policy browsing page enables you to edit all the parameters of a selected digital rights policy.

To edit a digital rights policy, select the digital rights policy from the browsing page and the click the **Edit** button. Clicking **Finish** saves the changes to the policy. Clicking **Cancel** sets the parameters back to their previous values and returns you to the browsing page.

Refer to Section 8.7.1 for descriptions of the parameters that you can edit.

8.7.3 Deleting a Digital Rights Policy

To delete a digital rights policy from the repository, select a policy from the browsing page and then click **Delete**.

8.7.4 Enabling or Disabling a Digital Rights Policy

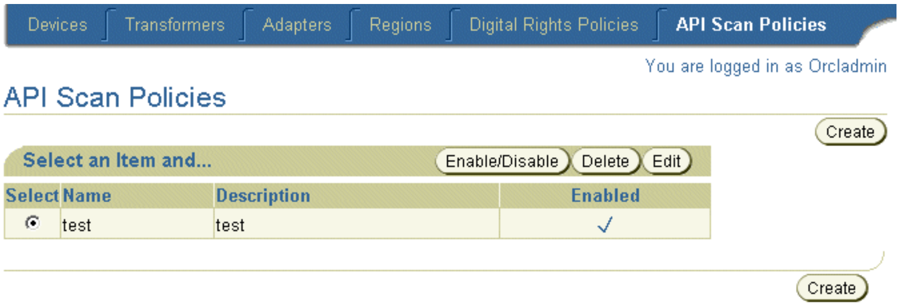
To enable or disable a digital rights policy from the repository, select a policy from the browsing page and then click **Enable/Disable**.

8.8 Managing API Scan Policies

An API scan policy defines the malicious APIs which can be invoked from a J2ME application that compromise a user's device. The API scan policy definition includes the malicious API package as well as the class and method names. During the certification process, the OracleAS Wireless server references the API scan policy objects when scanning a J2ME application for the APIs defined in the API scan policies. For information on how to scan a J2ME application, refer to Section 6.3.5.1.

You use the API Scan Policy subtab to manage the API scan policies. When you click the API Scan Policy subtab, the API scan policy browsing page appears (Figure 8–15), displaying a list of the API Scan Policies in the repository.

Figure 8–15 The Browsing Page for API Scan Policies



You are logged in as Orcladmin

API Scan Policies

Select an Item and...

Select	Name	Description	Enabled
●	test	test	✓

8.8.1 Creating an API Scan Policy

The API scan policy creation wizard enables you to create a policy. To access this wizard, click the **Create** button on the browsing page.

To define a policy, you must provide a name for the policy and then optionally enter a description.

Enter the XML Document which defines the malicious APIs. This XML document is based on the OracleAS Wireless Filter XML Schema. The text area of the *Create API*

Scan page displays a sample API scan document which defines the package, classes, and methods of the API that the OracleAS Wireless server references when scanning the J2ME application.

Click **Finish** to complete the API scan policy.

8.8.1.1 Editing an API Scan Policy

The **Edit** button in the API scan policy browsing page enables you to edit the description of a selected API scan policy.

8.8.1.2 Deleting an API Scan Policy

To delete an API scan policy from the repository, select the policy from the browsing page and then click **Delete**.

8.8.1.3 Enabling or Disabling an API Scan Policy

To enable or disable an API scan policy from the repository, select the policy and then click **Enable/Disable**.

Part III

Configuration and Integration

This section includes the following chapters:

9

OracleAS Wireless Gateway Configuration

This chapter describes how to configure OracleAS Wireless for voice and messaging communications through the following sections:

- Section 9.1, "Configuring Wireless for Voice Applications"
- Section 9.2, "Configuring OracleAS Wireless for Async-Enabled Applications"
- Section 9.3, "Configuring OracleAS Wireless for Notifications"
- Section 9.4, "Configuring Wireless for Browser-Based Applications"

9.1 Configuring Wireless for Voice Applications

Voice access is achieved through a VoiceXML gateway, a third-party server that connects to the telephony network on one side and OracleAS Wireless on the other. This voice gateway translates voice commands from the telephony line into HTTP requests which it sends to OracleAS Wireless. The voice gateway then renders the results as audio replies that are played back to the user.

After you install OracleAS Wireless, you can enable voice access. To do this, you must have the `Voice.ear` file (included with OracleAS Wireless) and obtain a third-party VoiceXML gateway that is approved by Oracle Corporation. OracleAS Wireless has been tested against a number of VoiceXML gateways. The list of accepted gateways is located at:

http://otn.oracle.com/tech/wireless/integration/content.html

Once you have selected a VoiceXML gateway provider, you then install and configure the gateway as instructed by the provider.

> **Tip:** See `http://www.oracle.com/technology/products/iaswe/OracleAS_Wireless_Voice_Deployment.pdf` for additional information on configuring and deploying voice access.

If you do not have access to a VoiceXML gateway, a number of gateway providers have hosted gateways for developers that can be used free of charge for development and testing purposes. For example, VoiceGenie maintains a developer studio at `http://developer.voicegenie.com`, where users can sign up for a development account that provides them with 10 extensions to a voice gateway. From the developer studio site, users configure each of their extensions to point to different URLs. To configure voice access to Wireless, you must set up an extension to point to the URL outlined in Section 9.1.1.

Note: To use the VoiceGenie hosted VoiceXML gateway for testing, you must edit the VoiceGenie device by adding the VoiceGenie*NXP/7.* user agent. For more information, see Section 8.3.

In addition to obtaining and configuring an approved VoiceXML gateway, you must do the following:

- Configure a Messaging Server voice driver (if needed), create an instance of the voice driver, and then add the instance to a Messaging Server process.
- Enter the number of the VoiceXML gateway as the value for the *Voice Access Phone Number* parameter in the *Basic Site Configuration* wizard (or in the *Voice Access Phone Number* parameter in the URL section of the *HTTP, HTTPS* configuration page of the System Manager). For more information, see Section 3.3.
- Enter the URL of OracleAS Wireless in the VoiceXML start page slot.

9.1.1 Provisioning Voice Access

To enable voice access, provision a VoiceXML gateway phone number to the following URL:

```
<server-name>:<port>/ptg/rm?PAlogin=true&PAlocale=<locale>
```

where the `port` is the WebCache listening port number, 7777. The default port number is 7777 and the port number range is 7777 to 7877.

You must specify the `locale` for a language other than English; if the locale is English, however, then you do not need the `PAlocale` attribute. Specify the locale using the two-letter Java locale format (the two-letter Java country code is optional). For example, to define the `PAlocale` attribute as French-Canadian, enter `fr_CA` (`fr` is the Java locale, `CA` is the country code).

This provisioning scheme contacts the voice login service for the OracleAS Wireless Server. After users login, they hear a main menu which lists all of the applications that they can access.

Note: Users must provide their account numbers and PINs to access the portal.

Use the `PAoid=<oid>` attribute to enable users to log into a particular application.

9.1.1.1 Provisioning Mobile Studio for Voice Access

When provisioning Mobile Studio for voice access:

1. Point the VoiceXML gateway to a URL for a start or login page in the Wireless and Voice Portal in the form of `http://<hostname>/ptg/rm`
2. Set the `PAlogin` parameter as `PAlogin=true`.

9.2 Configuring OracleAS Wireless for Async-Enabled Applications

This section provides an overview of configuring the e-mail and SMS access points that enable users to retrieve content through Asynchronous short name commands (also known as ASK commands) For more information, see Section 3.3 and Section 3.10.2.

Sections include:

- Section 9.2.1, "Enabling E-mail-Based (Two-Way Pager) Access"
- Section 9.2.2, "Enabling SMS Phone Access"

9.2.1 Enabling E-mail-Based (Two-Way Pager) Access

E-mail access to the OracleAS Wireless applications enables users to send an e-mail to a pre-defined address and then receive the requested content as a reply. In either the body or the subject line of this e-mail, the user enters a short name command to invoke one of the OracleAS Wireless applications, such as *stk <ticker symbol>* for information on a particular stock. OracleAS Wireless then replies with the requested data.

To enable e-mail access to wireless content:

1. Create an e-mail account for the incoming user requests.
2. Create site access point as described in Section 3.10.2.2.
3. Configure the appropriate Messaging Server driver instance. For example, to receive messages for a Async Listener e-mail address such as *foo@bar.com*, you must know the mail server which hosts the account, the protocol used (IMAP or POP3), and the user name and password. You must then create and configure an e-mail driver instance so that messages sent to *foo@bar.com* can be retrieved.

> **Note:** You must disable the e-mail receiving capability of the PushDriver. See Section 3.10.3.1.1.

4. Add the driver instance to a Messaging Server process.
5. Restart the Messaging Server process.

9.2.2 Enabling SMS Phone Access

Users retrieve content from the OracleAS Wireless applications by sending a text message containing the short name of the applications, such as *stk <ticker symbol>* described in Section 9.2.1. OracleAS Wireless replies with the requested content in a text message.

To deliver SMS messages, set up a communication channel to the SMS carrier that forwards the SMS content to OracleAS Wireless. You must contract with a carrier that has a network for sending and receiving SMS messages through the UCP or SMPP protocols and obtain SMS phone numbers as well. Alternatively, you can use an SMS aggregator, such as Mobile 365, which acts as an intermediary between the SMS carriers and the enterprise. (This may be beneficial when supporting messaging that requires multiple carriers.)

> **Tip:** For global SMS delivery, you must obtain an SLA from an SMS aggregator with a reasonable success rate in delivering messages worldwide. Unlike a single carrier, an SMS aggregator can offer expanded reach and coverage because of peering relationships and roaming partnerships. The SMS aggregator must be able to receive messages from global carriers and then communicate with OracleAS Wireless. A list of certified aggregators is located at:
>
> http://otn.oracle.com/products/iaswe/integration/content.html
>
> You also must create a driver that communicates with the aggregator's infrastructure.

To create an SMS access point:

1. Set up a telecom bridge that enables SMS delivery to the OracleAS Wireless over the wireless network. To do this, you must create a driver that communicates with an SMS aggregator's infrastructure. Before you contract with an aggregator, refer to the list of certified SMS aggregators located at:

   http://otn.oracle.com/products/iaswe/integration/content.html

   Alternatively, you can use one of the pre-built Messaging Server drivers that Support SMS or you can build a driver. If you build a driver, you must consult with the SMS aggregator and follow the OracleAS Wireless SDK. For more information on developing drivers, see to *Oracle Application Server Wireless Developer's Guide*.

   Once you have built a driver, you must add an instance of that driver to a Messaging Server process. For more information on creating driver instances based on Messaging Server drivers, see Section 3.10.3.1.

2. Create access points for the driver instance as described in Section 3.10.2.2. Enter a name for the access point, enter the SMS phone number as the address, select SMS as the delivery type, and select **Allowed to Access All Applications**.

9.3 Configuring OracleAS Wireless for Notifications

Notifications are delivered through different channels (SMS, e-mail, voice, or fax) and alert users to specific messaging events.

OracleAS Wireless ships with its *Messaging Server* component already configured to use the Oracle-hosted messaging gateway (`http://messenger.oracle.com/xms/webservices`). The Messaging Server communicates with hosted messaging gateway using the default Messaging Server driver, PushDriver.

Notes:

- Because PushDriver uses the HTTP protocol to communicate with the Oracle-hosted messaging gateway, you must therefore select the **Use Proxy** option in the OracleAS Wireless *Basic Site Configuration* wizard if you run the application behind a firewall. For more information see Section 3.3.
- If the OracleAS Wireless instance has been upgraded from the 9.0.2 to the 9.0.4 release, then the URL, `http://messenger.oracle.com/push/webservices` is valid (though the application will, in fact, be running on the 9.0.2 code base). As such, this URL is intended for backward compatibility only; update the clients of the SOAP API to use http://messenger.oracle.com/xms/webservices.

9.3.1 Configuring Non Oracle-Hosted Messaging Delivery

OracleAS Wireless ships with an account that enables 1000 units of notifications for SMS, voice, e-mail and fax messages. Because this account provides a limited number of notifications, you must obtain a certified messaging gateway provider, such as MutliMode, Inc., and then build a push driver appropriate to the messaging gateway provider. To contract with a certified messaging gateway provider, refer to

http://otn.oracle.com/products/iaswe/integration/content.html

Alternatively, you can configure the appropriate delivery channel for one of the OracleAS Wireless Messaging Server drivers that ship withOracleAS Wireless. OracleAS Wireless ships with 15 pre-built network drivers that support protocols that are accepted as industry standards. These drivers handle such communications protocols as SMS (short message for phone), e-mail (paging or desktop), voice and fax. To enable those network channels, you must configure the drivers to work with their corresponding network servers by identifying the external server to which OracleAS Wireless connects (this includes acquiring this connectivity and the configuration values for the Messaging Server drivers relevant to their particular protocols).

If you build a custom driver, you must consult with the service provider and follow the OracleAS Wireless SDK. For more information on developing drivers, see to *Oracle Application Server Wireless Developer's Guide*.

To enable those network channels, you must configure the pre-built drivers to work with their corresponding network servers. To do this:

1. Identify the external server to which OracleAS Wireless connects (this includes acquiring this connectivity and the configuration values for the OracleAS Wireless drivers relevant to their particular protocols).
2. Add the supported driver and configuring its messaging properties. This step is not required if you use the drivers that are packaged with the OracleAS Wireless Server.
3. Creating a Messaging Server process or selecting an existing one. This step is not required if you use an existing Messaging Server process.
4. Select the new Messaging Server process (or select an existing Messaging Server process) and create a driver instance for it. The driver instance properties must be configured to work with its corresponding external network connectivity. For details on driver configuration see Section 3.6.2.1

5. Re-starting the Messaging Server process. For more information on the Messaging Server process, see Section 3.6.2.

9.3.1.1 Configuring E-mail-Based Message Delivery

To configure the e-mail and paging services:

1. Set up an SMTP mail server for outgoing messages.
2. Optionally, set up an IMAP or POP3 mail server if message receiving is required.
3. Configure the e-mail driver and driver instance as described in Section 3.6.2.2.1 and Section 3.6.2.2.2, respectively.

9.3.1.2 Configuring the SMS Phone Message Delivery

To deliver SMS messages to phones, set up a communication channel to the SMS carrier. To do this, you must contract with a carrier that has a network for sending and receiving SMS messages through the UCP or SMPP protocols.

Alternatively, you can use a network aggregator, such as Mobile 365, which acts as an intermediary between the SMS carriers and the enterprise. (This may be beneficial when supporting messaging that requires multiple carriers.) Vendors whose protocols are certified to work with OracleAS Wireless are listed at:

http://otn.oracle.com/products/iaswe/integration/content.html

Configure the corresponding SMS drivers (for example: UCP, SMPP) and driver instances as described in described in Section 3.6.2.2.1 and Section 3.6.2.2.2, respectively.

9.3.1.3 Configuring Fax Delivery

For OracleAS Wireless Release 10*g* Release 2 (10.1.2.02), RightFax (a product of Captaris, Inc.) is the supported product for the delivery of fax messages. To enable delivery through the fax channel, you must acquire the RightFax product and follow its instructions to set up a fax server (described in Table 9–1). You must also use the Content Manager to edit the input parameters of the Fax mobile application. For information on editing applications (also known as application links), see Section 6.3.5.

Table 9–1 Required Software for the Fax Mobile Application

Name	Instructions	From Version
RightFaxServer (available from RightFax)	Install the RightFax server.	7.2

Table 9–1 (Cont.) Required Software for the Fax Mobile Application

Name	Instructions	From Version
RightFax Integration Application (available from RightFax)	Install the Integration Application on the Fax server.	7.2
RightFax PFD application (available from RightFax)	Install the PFD Application on the Fax server.	7.2

RightFax Java API (available from RightFax)	On Windows NT, copy `RFJavaInt.zip` (the Fax server's `RightFax/Production/xml/java` directory) to:
	`ORACLE_HOME%\wireless\lib`.
	On Solaris, copy `RFJAVAInt.zip` to:
	`ORACLE_HOME/wireless/lib`
	Include this `.zip` file in the OC4J classpath by adding the following line to `ORACLE_HOME\j2ee\OC4J_Wireless\config\application.xml`:
	`<library path=../../../wireless/lib/RFJava_api.zip"/>`
		7.2

The location of client API `.jar` files from RightFax must be added to the classpath in `ORACLE_HOME/wireless/sample/runpanamaserver.sh`. Configure the fax driver and driver instance as described in Section 3.6.2.2.1 and Section 3.6.2.2.2, respectively.

Sample Cover Page

Although the Fax mobile application uses a customized cover page file, you can use the sample cover page provided by OracleAS Wireless. To use this cover page, Microsoft Word 2000 must be installed on the RightFax server to enable server-side application conversion.

On Solaris installations, this cover page is located at:

```
ORACLE_HOME/j2ee/OC4J_
Wireless/applications/modules/modules-web/images/pim/fax/FCS.doc
```

On Windows NT installations, this cover page is located at:

```
ORACLE_HOME\j2ee\OC4J_
Wireless\applications\modules\modules-web\images\pim\fax\FCS.doc
```

To use the provided fax cover page:

1. Copy `FCS.doc` to the directory called `RightFax\FCS`, which is located on the machine on which you installed the RightFax server.
2. Specify the cover sheet.
3. Restart Enterprise Fax Manager.
4. Highlight **Users** under the appropriate server and double-click the user ID Administrator. Click the *Default Cover Sheets* tab. In the Cover Sheet Defaults group box, check **Send Cover Sheets** and select the cover sheet file (`FCS.doc`) in the *Cover Sheet Model* field.
5. Under the appropriate server, highlight **Groups** and then double-click the group **ID Everyone**. Click the *Basic Information* tab. Select the cover sheet file (`FCS.doc`) in the Cover Sheet Model field.
6. Restart Enterprise Fax Manager

Notes:

- See the RightFax administrator's guide for detailed instructions on fax cover sheets. You can download the documentation from:

  `http://www.captaris.com/rightfax/`

- The location of client API `.jar` files from RightFax must be added to the classpath in `$ORACLE_HOME/wireless/sample/runpanamaserver.sh`.

9.3.1.4 Configuring Voice Delivery

The voice driver implements the outbound telephony calls through a VoiceGenie VoiceXML Gateway. To configure the voice driver, provide the URL to the VoiceGenie Outbound Call servlet. Perform the remaining configuration of the voice driver and driver instance as described in Section 3.6.2.2.1 and Section 3.6.2.2.2, respectively.

Note: You must integrate with a VoiceGenie VoiceXML gateway for voice notifications. Provide the application URL to the VoiceGenie outbound call servlet (the required value for the `voicegenie.outbound.servlet.uri` parameter) by first selecting VoiceGenieDriver from the *Drivers* page and then by clicking **Edit**. From the *Driver Class and Parameters* section of the *Properties* page for VoiceGenieDriver, enter the URL for the `voicegenie.outbound.servlet.uri` parameter.

9.4 Configuring Wireless for Browser-Based Applications

This section describes how to configure OracleAS Wireless for PocketPCS, Palm, and WAP phone applications. Topics include:

- Section 9.4.1, "Configuring OracleAS Wireless for PocketPCs"
- Section 9.4.2, "Configuring OracleAS Wireless for PALM"
- Section 9.4.3, "Configuring OracleAS Wireless for WAP"

9.4.1 Configuring OracleAS Wireless for PocketPCs

This section describes the procedures for configuring Oracle Application Server Wireless to PocketPCS. Topics include:

- Section 9.4.1.1, "Connecting to the Network"
- Section 9.4.1.2, "Accessing the OracleAS Wireless Server Using Internet Explorer"
- Section 9.4.1.3, "Setting Up the Internet Explorer Home Page"

9.4.1.1 Connecting to the Network

To access the OracleAS Wirelesss server from a Pocket PC device, connect the device to the network. If the OracleAS Wireless server is on a corporate Intranet, then you must connect the device to the corporate Local Area Network (LAN). If the Wireless server is on the Internet, then you must connect to the Internet Service Provider (ISP). The other methods for connecting the Pocket PC device to a corporate LAN or ISP are documented in the Pocket PC Connection Manager tutorial at:

http://www.microsoft.com/mobile/pocketpc/tutorials/connectionmanager/default.asp

9.4.1.2 Accessing the OracleAS Wireless Server Using Internet Explorer

To access the OracleAS Wireless server using Internet Explorer:

1. Open Internet Explorer by clicking **Start** in the desktop, then by selecting Internet Explorer. (If you are already in Internet Explorer, go to Step 2).
2. Select **View**, and then **Address Bar** to display the Internet Explorer Address Bar (If the Internet Explorer Address Bar displays, go to Step 3.)
3. Enter the URL to the OracleAS Wireless server in the Address Bar and click the **GO** button (represented as a green arrow).

9.4.1.3 Setting Up the Internet Explorer Home Page

Once connected to the main page on the OracleAS Wireless server, you can make that page the home page for Internet Explorer so that you do not have to repeatedly enter the URL to OracleAS Wireless.

1. While still displaying the OracleAS Wireless server main page select **Tools**, then **Options....**
2. Select the **Use Current** button in the *Home* page section.
3. Select **OK**.

9.4.2 Configuring OracleAS Wireless for PALM

There are two types of Palm devices for connecting to the Internet and Intranet:

- Devices with built-in wireless Internet access (Palm i705)
- Devices that require an Internet Service Provider (ISP) account and data-enabled phone or modem to access the Internet (Palm m515, Palm m505, Palm m500, Palm m130, Palm m125).

For a device with built-in wireless Internet access, you need only to activate the wireless service to connect the device to the Internet.

For devices that do not have built-in Internet access require an ISP account and either a data-enabled phone or a Palm modem. (A data-enabled phone or a Palm modem will suffice for the Palm i705l.)

9.4.2.1 Configuring the Connection Method

To configure the connection method:

1. Open Preferences by clicking the **Press** icon.
2. Select **Connection**.
3. Select the connection method from the list of Available Connections.

9.4.2.2 Configuring an ISP Account

To configure the ISP account.

1. Open Preferences by clicking the **Press** icon.
2. Select **Network**.
3. Select the service value from the drop down list.

4. Enter the user name.
5. Enter the password.
6. Select the connection type from the drop-down list.
7. Enter the phone number.
8. Click the **Connect** button to test the settings.

If the handheld device supports more than one-way to connect to the Internet, you can use any preferred method.

1. Open Preferences by clicking the **Prefs** icon.
2. Select **Web Clipping**.
3. Select the connection name from the drop down list.

9.4.2.3 Accessing the Wireless Server Using MyPalm Application

If you have a Palm device with built-in wireless Internet access and you have activated the wireless service, then you can use the Palm native web browser to access a wireless server.

1. Open MyPalm application by clicking the *MyPalm* icon.
2. Enter the URL to the wireless server and click the **Go** button.

9.4.2.4 Installing Blazer Web Browser

To install PalmOne's Blazer browser:

1. Download the Blazer browser software from http://www.palmone.com
2. Follow the installation instructions provided at:

   http://www.palmone.com.

9.4.2.5 Accessing the Wireless Server Using Blazer

1. Open Blazer by clicking the **Blazer** icon.
2. Click the **Go to Web Page** icon (the opened folder icon).
3. Enter the URL to the OracleAS Wireless server and click the **OK** button.

> **Tip:** Create a bookmark so that you do not need to repeatedly enter the URL.

9.4.3 Configuring OracleAS Wireless for WAP

The OracleAS Wireless server does not directly not support the WAP protocol, which enables WAP devices to communicate. As a result, supporting WAP requires a WAP gateway to convert the WAP protocol to HTTP(S). If you can connect to the Internet through the wireless service provider, then the provider has already configured a WAP gateway for you. However, if you connect to the Internet (or Intranet) through a dial-up (PPP) connection, then you must install and configure a WAP gateway.

9.4.3.1 Configuring a WAP Phone

The WAP phone configuration is specific both to the phone model and to the wireless service provider. In general, the phone must be configured for a dial-up network connection (this does not apply to GPRS phones), the WAP gateway, and the home URL for the WAP browser.

Generally, the phone is configured by the wireless service provider to connect to their own WAP gateway. Some wireless service providers hide the phone settings to prevent the user from changing them. In most cases, you do not need to change the phone network settings; instead, to access the wireless server from a WAP phone, you need only enter the URL of the wireless server into the phone’s WAP browser. (Refer to the phone’s user's manual for instructions on opening the WAP browser.)

OracleAS Wireless serves requests from different devices, including Palm, Pocket PC, and WAP. These devices must be configured so that they can access the OracleAS Wireless server. Requests from these devices to the wireless server come through an HTTP(S) protocol transformation gateway may be used in some cases to convert the device native network protocol to HTTP(S).

Note: The URL to the OracleAS Wireless server must be configured for all devices. If the OracleAS Wireless server is installed on the host (*host.domain*), then the default URL for HTTP and HTTPS protocols are:

- `http://host.domain:7777/ptg/rm`
- `https://host.domain:4443/ptg/rm`

Consult with an OracleAS Wireless server administrator for the exact URL to the OracleAS Wireless server.

10

OracleAS Wireless Security

This chapter includes the following:

10.1 Overview of OracleAS Wireless Security

OracleAS Wireless combines advanced content transformation, device adaptation and network adaptation services with end-user customization, providing enterprises, mobile operators, content providers, or wireless ISPs with a platform to create and deploy mobile applications. OracleAS Wireless incorporates various security mechanisms that enable the deployment of end-to-end secure, unbreakable applications.

To provide a clear understanding of security and its application in the wireless world, this section provides brief descriptions of the principles of security and describes common application deployment models for both the wired and wireless world, explaining their similarities and differences in regards to security. Subsequent sections describe these security principles in more detail, provide available deployment scenarios, and identify any issues that are wireless-specific and the present OracleAS Wireless solution.

The principles of security are:

- **Communication Data Privacy**: Unintended parties cannot observe data during transmission (on the network).
  - Data Privacy usually denotes encryption of data, either at the transport layer or at the application layer.
  - Technologies for communication data privacy are Virtual Private Networks (VPNs) and secure transport layer protocols (for example, WTLS, TLS).
- **Authentication**: Verifying the identity of one or more parties (that is, *who is the user?*).
  - Authentication denotes a wide range of technologies with different requirements and degrees of security, including user names and passwords, certificate-based 2- (or 3-) factor authentication.

- **Authorization**: Access control of authenticated parties (that is, what can the user do?).
  - Authorization involves checking bindings between user identities with user capabilities: "what is this user allowed to do?"
  - Most authorization systems involve the concepts of Users, Groups, Roles, Policies and Access Control Lists (ACLs).
- **Data Integrity**: Data cannot be tampered with when in transit or in storage.
  - Data Integrity means protection from malicious or accidental data alteration, data omission and data replay (that is, avoid replay attacks).
  - Several technologies provide data integrity in such forms as Message Authentication Codes (MACs), digital signatures, protection through encryption.
- **Non-Repudiation**: Authenticated users cannot disclaim the transactions that they have made.
  - Non-repudiation allows for digital content signing and enables contract enforcement by making transactions undeniable and openly verifiable (that is, verifiable to a third party).
  - Non-repudiation is usually achieved using digital signatures.
- **Storage Data Privacy**: Unintended parties cannot observe sensitive data (for example, credit card numbers) during storage (on the database or file system).
  - Storage Data Privacy usually denotes a combination of access controls and encryption of highly sensitive data.
  - AES (Advanced Encryption Standard) is the new symmetric encryption algorithm approved by the U.S. Federal Information Processing Standard (FIPS).
- **Accountability**: As part of accountability, auditing enables logging of security-related traces for all other security principles.
- **Availability**: Includes attack countermeasures to protect the system from attacks such as denial of service attacks.

Note: This chapter discusses the security principles which are specific to OracleAS Wireless (Communication Data Privacy, Authentication, Authorization and Non-Repudiation). Beyond this overview, this chapter does not describe the non-OracleAS Wireless principles, such as Storage Data Privacy.

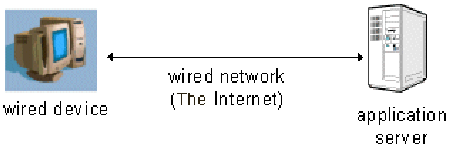
10.1.1 Wireless Security and Wired Security: A Comparison

This section describes the differences and similarities of security in both wired deployment and wireless deployment.

10.1.1.1 Wired Application Deployment

Figure 10–1 depicts the basic arrangement of deployment in the wired world: a wired device, such as a PC, connects over the network to an application server.

Figure 10–1 Wired Deployment



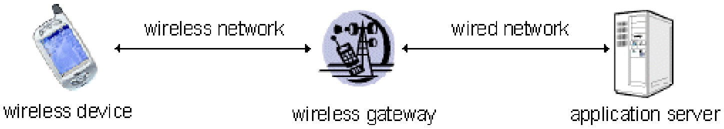
The main security characteristics of the wired deployment scenario are:

- Data travels across the wire. This data may be protected by a secure communication protocol, such as SSL. Encrypted communication between the device and the application server is regarded as end-to-end secure, as the communication can be carried out without intermediate nodes that intercept and modify the information.
- Online application access is usually controlled through user name and password authentication (traveling over the protected communication link). More secure schemes make use of digital certificate-based technology or tokens (for example, RSA SecurIDs).
- Access control is carried out at the application server side by checking the permissions set for the authenticated user.
- Data integrity is provided along with communication data privacy through encryption.
- Wired applications requiring some measure of non-repudiation usually resort to using transaction logs. Strong non-repudiation can be carried out through digital signatures.
- Sensitive data residing at the server side in the database can be protected through encryption and access controls.
- Log files provide security auditing of transactions and malicious activity.
- Attack countermeasures, which usually include fire walls and demilitarized zones (DMZs), restrict direct application server exposure to the public network (that is, the Internet).

10.1.1.2 Wireless Application Deployment

Figure 10–2 depicts the deployment scenario for wireless devices. The wireless device, a device limited in both power and bandwidth, stands at one end of the transaction. The wireless device communicates over the air through a wireless network to a gateway component which performs the translation from the wireless network protocol to the wired network protocol so that the device can contact the application server.

Figure 10–2 Security Chain in Wireless Transaction Flow



The wired side of the wireless network is practically the same as the wired deployment scenario explained in Section 10.1.1.1. However, because of the added components in the transaction flow, the following security considerations arise:

- **Network protocol conversions**: In contrast to the wired deployment scenario, wireless application deployment requires that the wireless gateway intervenes in the communication to perform protocol conversions from the wireless network to the wired network. The problem arises when the wireless protocol is not directly interoperable with the wired protocol (as it is in many cases), causing network level communications to no longer be end-to-end secure and thus becoming only point-to-point secure. Such "leg-based" communication may be justifiable where there is need for little or no security (for example, a public news server) but may not be acceptable for the most security-conscious applications, such as mobile banking or corporate applications.
- **Limited computational power and bandwidth in the wireless network and device**: The restricted power of the wireless device along with the low bandwidth of wireless networks requires the deployment of more efficient and economical encryption mechanisms such as Elliptic Curve Cryptography (ECC) in WPKI. This requires special support by the application server in terms of integration.
- **Lack of well-defined authentication standards**: While password authentication is both common and standard in the wired world, it is not perceived as being highly secure, especially in the context of mobile applications. This causes the introduction of various authentication mechanisms with tight coupling to the physical wireless device causing authentication (and other types of security such as non-repudiation) mechanisms to be dependent on the device.

10.1.2 Classes of Users and Their Privileges

There are two classes of OracleAS Wireless users: registered users and anonymous users. The registered users are users whose user information is registered with the Oracle Internet Directory (OID). These users can be created, modified, or deleted through the User Manger or the OID DAS tool. Anonymous users are these users that have not been registered with OID. Anonymous users can only access the wireless and voice applications assigned to the Guest group. A registered user can only access the wireless and voice applications assigned to the groups to which that user belongs. For more information on anonymous users and assigning users to groups, see Chapter 5, "Managing Users". For information on assigning applications to user groups, see Section 6.5. For more information on creating users with OID, see the *Oracle Internet Directory Administrator's Guide*.

The OracleAS Wireless Tools, such as the User Manager, are role-specific; OracleAS Wireless users can only access the tool which corresponds to the role or roles that they have been granted. The User Manager assigns these roles when creating (or updating) a user. A user can have one or several roles, which include System Administrator, Application Developer, Foundation Developer, Content Manager, User Administrator, and End User. These roles span all of the OracleAS Wireless resources, from server management, application development, application publishing, and help desk to subscription to the OracleAS Wireless applications. For more information on OracleAS Wireless user roles, see Section 10.2.

10.2 Resources Protected by Oracle Application Server Wireless

The OracleAS Wireless meta data repository does not store any sensitive information. Instead, information such as the user passwords, voice-accessed PINs, and the password to the OracleAS Wireless meta data schema are stored in Oracle OID.

Sensitive resources (such as the OracleAS Wireless Tools) are protected through access controls and various authentication mechanisms, such as user names and passwords. Service access is also protected the user names and passwords.

10.2.1 Authorization and Access Enforcement

Access to the OracleAS Wireless tools is controlled through user roles, which not only provide access to the tools, but define the capabilities of the OracleAS Wireless user as well. Table 10–1 describes the user roles, their capabilities and the resources that these roles enable.

Table 10–1 Wireless User Roles

User Role	Description	Available Tools
Application Developer	Users assigned the Application Developer role perform the following functions: ■ Create, modify, delete and test applications. ■ Publish applications to the Application Developer's folder. ■ Create, modify, and delete notifications. ■ Create, modify, and delete data feeders. ■ Register and delete J2ME Web services. ■ Develop preset definitions.	Service Manager
Foundation Developer	Users assigned the Foundation Developer role perform the following functions: ■ Create, modify, and delete devices. ■ Create, modify, and delete transformers. ■ Create, modify, and delete regions. ■ Create, modify, and delete digital rights policies. ■ Create, modify, and delete API scan policies.	Foundation Manager
Content Manager	Users assigned the Content Manager role perform the following functions: ■ Manage application folders and bookmarks. ■ Create application links based on Application Developer-created applications. ■ Create notifications based on alerts (deprecated in this release). ■ Create application categories and associate access points with them. ■ Create a user-home folder rendering scheme, such as setting the sorting order for applications.	Content Manager
System Administrator	Users assigned the System role manage the system using the System Management Tool.	Wireless system management functions (through the Oracle Enterprise Manager Application Server Control).

Table 10–1 (Cont.) Wireless User Roles

User Role	Description	Available Tools
User Manager	Users assigned the User Manager role perform the following functions: ■ Manage users by providing such Help Desk functions as editing a user profile, resetting passwords and PINs, and creating or deleting users. ■ Manage user access privileges. ■ View application links assigned to users. ■ Manage user devices. ■ Search for users. ■ View overview information of users.	User Manager
Sensor Services Administrator	User assigned the Oracle SensorEdge Services Administrator role manage the devices and filters used with the Edge Server. These functions inlcude: ■ Create drivers for Oracle Sensor Edge Server devices ■ Create filters used with Oracle Sensor Edge Server devices ■ Manage the Oracle Sensor Edge Server devices assigned to the Edge Server processes	Sensor Services Tool
End User	Users assigned the end user role are the consumers of OracleAS Wireless services. End-users create their own accounts when they register with OracleAS Wireless using the OracleAS Wireless Customization. End users can also customize their own services either from a desktop or from a device. Customization for end-users includes: ■ Customize applications, download J2ME applications, subscribe to notifications. ■ Manage devices. ■ Manage location marks and location settings. ■ Manage contact rules. Mobile studio users also have the end user role; a user belonging to the StudioUser group can access the Mobile Studio. Every OracleAS Wireless user is granted the Mobile Customer Role by default. This role is implicit to all users.	Wireless Customization Portal Mobile Studio (for users assigned to the StudioUser group)

In OracleAS Wireless, a user group is the means by which users can access any voice and wireless application; any application that has been published to a user group is available to all of that group's members. The Content Manager can both create a user group and assign applications to a user group, which is a collection of users. The user manager assigns users to user groups. See Section 6.5 for information on publishing an application to a user group and Section 6.5 for assigning a user to a user group.

10.2.2 Authentication Through User Names and Passwords

Access Control to applications in Oracle Application Server Wireless is provided according to the channel used to connect to the server. For visual HTTP-based channels such as WAP, Oracle Application Server Wireless authenticates users through user names and passwords; for voice-accessed applications, OracleAS Wireless uses account numbers and PINs; for message-related applications, OracleAS Wireless checks the user account information (for example, OracleAS Wireless checks the e-mail headers). For web services related to the messaging infrastructure, OracleAS Wireless authenticates users through user names and passwords.

10.2.3 Device-Based Authentication Mechanisms

Besides user name and passwords, OracleAS Wireless allows for other authentication mechanisms, depending on the device. However, the application developer is responsible for choosing and integrating the appropriate mechanism for the target device. The authentication mechanisms available in the various channels and how they can be used are as follows:

- **WAP**: With WPKI, the end users can utilize their WAP device to sign a "challenge" (a randomly generated string) sent by an authentication service. The authentication service requests the signature through WMLScript's `signtext()` function. Upon receiving a signature request, the WAP device prompts the user to enter his or her local PIN (which authenticates the user only to the WAP device) and then retrieves the user's WPKI private key (stored in a SIM chip in the WAP phone) to sign the challenge. Once the authentication service receives the signed challenge, it is verified with the user's WPKI public key (possibly stored in a user's certificate repository such as the Oracle Wallet Manager) and notifies the requesting application of the result. WMLScript's `signtext()` function is available with WAP 2.0.
- **SMS**: SMS-based authentication can occur in two ways with varying degrees of security. The most basic authentication is to reply (with a PIN) to an SMS received from the authentication service. Another SMS-based authentication mechanism relies on digital signatures; this mechanism is similar to the WAP case.
- **E-mail**: Authentication consists of sending a reply (with a PIN) to an e-mail sent by the authentication service.
- **Voice**: The authentication service calls the user and asks for a PIN. Once the user provides the PIN, the system then authenticates the user.

10.2.4 How Oracle Application Server Wireless Leverages Security Services

The OracleAS Wireless tools and the Customization Portal are protected by the Oracle HTTP Server SSO (Single Sign-On) plugin module (mod_osso). The mod_sso protects all the URL access to the OracleAS Wireless tools. If any part of the URL access is not authenticated, then the mod_sso redirects the request to SSO for authentication. For more information, see the `Oracle Application Server Single Sign-On Administration Guide.`

To further secure the communication channel between the browser and the OracleAS Wireless tools, or the wireless gateway (for example, the WAP gateway, or the voice gateway), you can enable SSL on the Oracle HTTP Server. For more information on configuring SSL on the Oracle HTTP Server, see the `Oracle HTTP Server Administrator’s Guide.`

In addition, you can also enable the SSL-based secured communication channel between the OracleAS Wireless Multi-Channel Server and remote application server by installing either Base64 certificate or PKCS#7 formatted certificate at the OracleAS Wireless Multi-Channel Server. You can install such a certificate through the System Manager (accessed through Oracle Enterprise Manager). For information on using the System Manager to configure an SSL certificate, see Section 3.9.1.

SSO is a feature (one not specific to OracleAS Wireless) that eliminates the need for repeated authentication (within a period of time) when crossing application boundaries for the same trusted domain. It also provides for centralized user credentials, which avoids the problem of having to remember passwords for different applications, thereby increasing security for the whole system, as passwords do not need to be written down.

OracleAS Wireless is fully integrated with SSO, which currently supports authentication through user names and passwords over all visual HTTP-based channels and through account number and PIN for voice-based channels.

SSO also integrates with the Oracle Internet Directory (OID), an LDAP server that stores, among other things, valid end-user authentication information such as passwords and digital certificates.

The user information stored in OID is replicated to the OracleAS Wireless meta data schema when a user logs in, or through asynchronous synchronization from OID to the OracleAS Wireless schema. Table 10–2 lists the user attributes (stored in OID) that are replicated in OracleAS Wireless schema.

Table 10–2 User Attributes Stored in the OracleAS Wireless Schema

Attribute Name	Description
orclCommonNickNameAttribute	The user name used for authentication for all channels, except voice. By default this is *cn* (as specified in OID configuration).
userPassword	The user password, used for authentication for all channels, except voice.
orclPasswordHint	The password hint
orclPasswordHintAnswer	The answer to the password hint
orclWirelessAccountNumber	The account number, used for authentication from voice channel. This must be comprised of digits only.
orclPasswordVerifier; orclCommonPIN	The PIN used for voice authentication. This must be comprised of digits only.
displayName	The display name of the user
orclIsEnabled	A flag whether the user is enabled
preferredLanguage	The Locale, such as the. language and country, for example en_US indicates English and USA
orclTimeZone	The user's default time zone
orclDateOfBirth	The user's date of birth
orclGender	The user's gender

10.2.5 Component Extensibility and Security

Applications developed and deployed in OracleAS Wireless can benefit from Oracle SSO functionality through integration as an Oracle SSO partner.

10.3 Configuring the Security Infrastructure to Support Wireless

OracleAS Wireless depends on the security infrastructure to be up both during installation time and runtime. Refer to the Oracle Application Server Administrator's guide for details on the security infrastructure.

OracleAS Wireless relies on Directory Integration Platform (DIP) server, as one of the mechanisms, to asynchronously replicate the essential modified user information from OID to the OracleAS Wireless schema. Refer to the *Oracle Internet Directory Administrator's Guide* for details on how to start the DIP server.

By default, the OID server does not enforce unique constraints on account number (that is, the `orclWirelessAccountNumber` attribute of `orclUserV2` object class). The account number is required for users accessing wireless applications from a regular voice line with the account number and PIN used for the authentication. As part of the OracleAS Wireless installation, the OracleAS Wireless configuration assistant enables the policy to enforce a unique constraint on the `orclWirelessAccountNumber` attribute of `orcluserV2` object class. The OID server must be restarted after the first OracleAS Wireless installation for this unique constraint policy to take effect. Refer to Oracle Internet Directory Administrator's guide for details on how to restart the OID server.

OracleAS Wireless connects to the OID as a OracleAS Wireless application entity after users have been authenticated through SSO. The OracleAS Wireless application entity is assigned following privileges:

1. **Common user attributes**: privilege to read common attributes of a user.
2. **`OracleDASCreateUser`**: The privilege to create users in OID.
3. **`OracleDASDeleteUser`**: The privilege to delete users in OID.
4. **`OracleDASEditUser`**: The privilege to edit common attributes of users.
5. **`verifierServices`**: The privilege to read application verifiers (the user PIN) which are stored in the user.
6. **`authenticationServices`**: The privilege to perform compare operations on password attributes of a user.

By default, the OracleAS Wireless application entity does not have the privileges to change the user password. Consequently, out-of-the-box users cannot change their password from theOracleAS Wireless server. However you can enable the functionality to change passwords by assigning the `UserSecurityAdmins` privilege to the OracleAS Wireless application entity. To do this, execute *assignUserSecurityAdminsPrivilege.sh* (or *assignUserSecurityAdminsPrivilege.bat*, depending on the operating system) on the machine on which OracleAS Wireless is installed. The script is available in the ORACLE_HOME/wireless/bin directory.

The syntax for invoking the utility is as follows:

```
assignUserSecurityAdminsPrivilege.sh oid_super_user_dn user_password
```

where

`oid_super_user_dn` is the Distinguished Name (DN) of the OID super user. The user should have privileges to grant `UserSecurityAdmins` privilege to application entities

`user_passord` is the password of the OID super user

For example:

```
assignUserSecurityAdminsPrivilege.sh cn=orcladmin welcome1
```

10.4 Installing and Configuring Oracle Application Server Wireless Security

This section describes Communication Data Privacy and Non-Repudiation principles of wireless security. In discussing these principles, this section provides alternatives available to application developers when incorporating security in OracleAS Wireless. In some cases, OracleAS Wireless does not provide direct support for security on a given channel, leaving the responsibility to the application developer to recognize the security needed for the application and to deploy the appropriate security mechanisms.

10.4.1 Communication Data Privacy

Communication Data Privacy is the principle of security that prevents data in transit (on the network) from being partially or completely observed by unintended parties or eavesdroppers. Along with authentication, communication data privacy is one of the most important aspects of wireless application security.

This section focuses on end-to-end data privacy, where no intermediate nodes (or actors) are able to understand the data that passes through them. For example, the wireless carrier's WAP gateway should not be able to understand the sensitive information, even when all the data passes through it. End-to-end data privacy stands in contrast to point-to-point (or leg-based) data privacy, where data is secured between servers and devices but intermediate nodes can see the data in "the clear".

10.4.2 Data Privacy Deployment Solutions

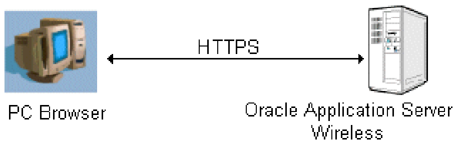
Given the variety of (wireless) networks and protocols, communication data privacy differs among the various communication channels. The following sections describe communication data privacy deployment solutions based on the communication channels:

- Section 10.4.2.1, "PC Browsers"
- Section 10.4.2.2, "Pocket PCs"
- Section 10.4.2.3, "Short Messaging Service"
- Section 10.4.2.4, "E-mail"
- Section 10.4.2.5, "Voice"

10.4.2.1 PC Browsers

Internet Protocol (Web) communication is currently used today in the wired world with standardized security through SSL encryption. PCs connect to the application server directly with point-to-point privacy using HTTPS, which is HTTP running over an SSL-secured link (Figure 10–3). Since there are no intermediate nodes performing protocol translations at the security layer (as there are WAP 1.x), data communication is end-to-end private between the browser and the application server.

Figure 10–3 PCs Browsers

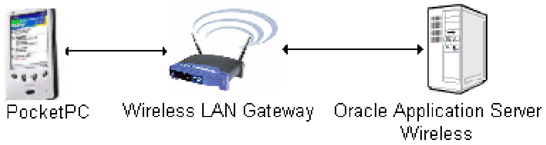


Because OracleAS Wireless supports HTTPS connections, PC browser connections are point-to-point secure.

10.4.2.2 Pocket PCs

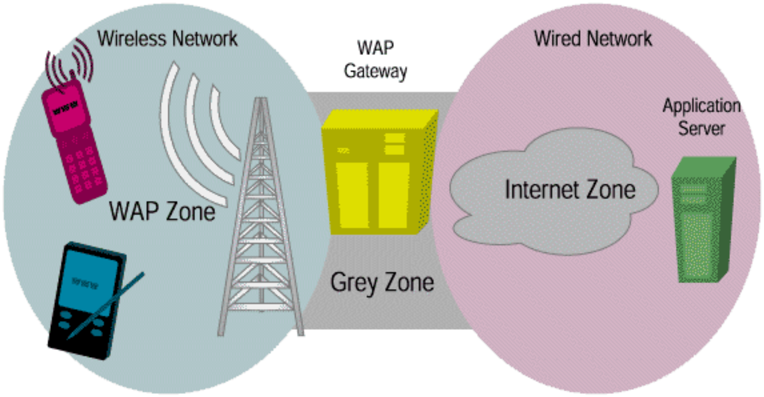
The wireless extension to the PC browser is through the use of HTTP devices that connect to a wireless LAN gateway, as it is in the case of Pocket PCs with wireless LAN card adapter. The connection from the device to the wireless LAN gateway follows the 802.11b standard for wireless communication, which is interoperable with the wired Internet protocol since it uses the Ethernet protocol. Since both protocols (wired and wireless) are interoperable at the security layer, point-to-point secure communication is also carried out through HTTPS from the device to the Application Server (as depicted in Figure 10–4).

Figure 10–4 Wireless LANs and Other HTTP-Based Devices



Without HTTPS security at the application layer, wireless LANs are insecure even with the use of the Wired Equivalent Privacy (WEP) protocol, a protocol operating at the data link layer designed to protect communication but which has been shown to be insecure. Therefore, wireless networks that depend solely on WEP for privacy are found to be vulnerable to "war-driving", an attack where the eavesdropper 'drives by' with a wireless receiver to break WEP security and decode wireless information.

Figure 10–5 Wireless Application Protocol (WAP)



Security in the Wireless Application Protocol (WAP) is currently specified in the WTLS (Wireless Transport Layer Security) protocol. Similar in design to SSL (TLS), but optimized for bandwidth and power, WTLS provides privacy from the wireless device to the WAP gateway, allowing for server authentication and mutual authentication modes.

WAP has been widely criticized by the security sector on what is commonly called the 'WAP gap', which breaks end-to-end communication data privacy. The WAP device communicates with the WAP gateway through WTLS (the WAP pictured in Figure 10–5) and the WAP gateway, in turn, communicates with the application server using SSL (Internet zone.) Since WTLS is not compatible with SSL because of handshake optimizations, a protocol translation needs to occur at the WAP gateway (the Grey Zone pictured in Figure 10–5): that is, WTLS-encrypted data must be decrypted and be SSL-encrypted. The 'WAP gap' refers to this split second when the data is in the clear at the WAP gateway; this alone breaks end-to-end privacy and is a cause of concern for the banking industry and the most security-conscious.

For WAP 1.x deployments, bridging the WAP gap can be accomplished by redirection to subordinate pull proxy (gateway) with WAP 1.2. If, in addition to the WAP gateway at the carrier side, there is another WAP gateway residing within the same physically secured, trusted domain as the application server (that is, both are owned by the same company), then communication can be redirected to this enterprise gateway and can thus be considered end-to-end private. In this way, the WTLS connection would be established with a gateway located at the site of the application service provider and it would also allow for WTLS class 3 (client and server authentication).

Some gateways, such as the OpenWave's Mobile Gateway server, already support the deployment of proxy gateways at the premises of the content provider. In this model, after the subordinate proxy discovery process, requests will be rerouted to the proxy gateway installed within the secure premises of the provider hosting the OracleAS Wireless application server. The network operator retains the control of the calls at the cost of sharing the burden of supporting the infrastructure required for end-to-end secure communications.

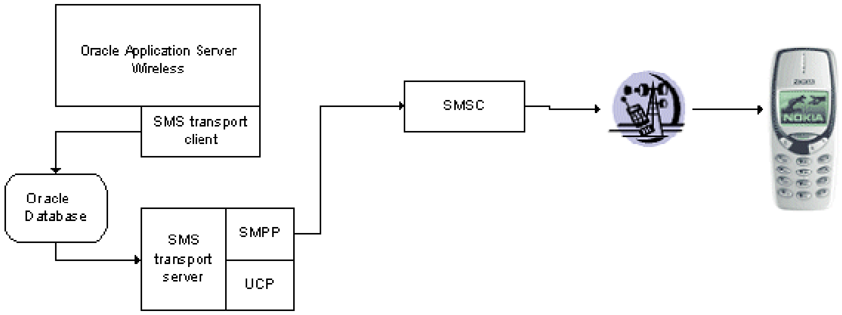
The disadvantage of this solution is that the company hosting the application server must deploy and maintain its own WAP gateway. User Agent devices, such as Nokia Mobile Browser 3.0, already support this deployment model.

In the next generation of WAP (WAP 2.0), WAP designers will eliminate WAP gap by introducing Internet Protocols. That is, WAP devices which can securely communicate using SSL. In addition, no translation will be necessary at the WAP gateway, thus providing end-to-end privacy. This is a step to ensure that WAP devices are interoperable with the wired Internet.

10.4.2.3 Short Messaging Service

The Short Message Service (SMS) deployment architecture dictates that messages be routed through a Short Message Service Center (SMSC) from the application server to the wireless device (as depicted in Figure 10–6).

Figure 10–6 Short Message Service (SMS)



Under SMS, required security for a given deployment scenario depends largely on the business model (for example, carriers versus corporations) of the enterprise deploying the solution. Given these different business models, secure deployment alternatives are as follows:

1. **No Transport Security Needed**: This scenario relies on the existing security of the wireless network protocol provided from the SMSC to the wireless device (for example, GSM network security.) In this scenario, there is no secure link between the application server and the SMSC. This deployment alternative is end-to-end secure only when the application server and the SMSC reside within the same secure domain (that is, both SMSC and application server are co-located in the same physically secured zone to reduce risk from internal eavesdroppers or attackers); carriers providing applications to their subscribers benefit the most from this solution as it fits their business model.

2. **Point-to-point security**: Another alternative consists in securing the link between the application server and the SMSC with the use of VPN (virtual private network) or SSL-secured connections. This deployment alternative applies when the application server and the SMSC reside in different (albeit secure) domains. Unfortunately, a problem similar to the 'WAP gap' (see Section 10.4.2.2) occurs here because there is a translation from the wireless protocol (used for communication between the wireless device and the SMSC) to the wired protocol (used for the communication between the SMSC and the application server) that leaves the data exposed at the SMSC. In other words, this deployment solution is not considered end-to-end secure. However, given current technology, this is the optimum deployment scenario for businesses that do not have a pre-existing relationship with their customers, as is the case for merchants.

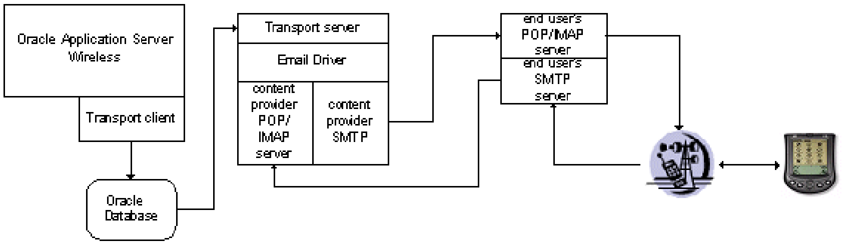
3. **Application Level security with Symmetric Shared Encryption Keys**: This deployment scenario provides end-to-end data privacy by performing symmetric encryption at the device and at the application server: since data is encrypted above the network protocol, no intermediate nodes can observe the data while en route. End-to-end communication data privacy under this deployment scenario requires application level encryption support through a SIM/WIM card with encryption capabilities. OracleAS Wireless currently supports Triple DES symmetric key encryption at the application layer. This means that tag content information (that is, not all of the payload) is encrypted and decrypted at the device and is then decrypted and encrypted at the application server.and that there is a shared secret encryption key between each user and the OracleAS Wireless application server. The encryption key is stored in the SIM card and is initially produced at time of manufacture and re-keying on a periodic (or other) basis is possible under a set of well-defined security conditions. This deployment scenario best fits corporations that want to provide applications to their mobile field forces.

   A more scalable and generic alternative is the use of application layer PKI encryption, which eliminates the need of a pre-existing relationship between end-consumer and the business. Unfortunately, there are currently no SIM card vendors that offer PKI encryption capabilities (only signature capabilities) as there is no standardized way for key generation and certificate provisioning for SMS.

10.4.2.4 E-mail

Just as in SMS, end-to-end private e-mail communication depends on the deployment scenario.

Figure 10–7 E-mail



Depending on such factors as the capabilities of the e-mail device and the location of the end-user e-mail server in respect to the application server, several alternative solutions exist:

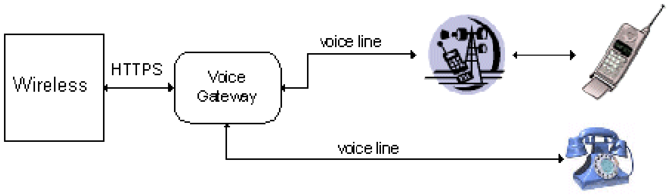
- **No Transport Security Needed**: In this deployment scenario, an end-user's e-mail server and the application server both reside within the carrier's secure domain. Since e-mail communication never leaves the secure domain (that is, the only direction of communication is from within the carrier to the device), there is no need to additionally secure the communication beyond the security provided by the wireless network protocol in the "air." The disadvantage of this solution is that it applies only to carriers that offer at the same time both e-mail and application services to their subscribers.
- **Point-to-point security**: In this deployment scenario, the e-mail server and the application server both reside within the same secure domain at an enterprise other than the carrier. This solution secures communication from the carrier to the enterprise e-mail server by establishing a TLS- or SSL-secured connection such as Secure SMTP. In this scenario, the wireless device uses the carrier's system to retrieve the e-mail message, and in turn the carrier system communicates with the e-mail server using a TLS or SSL secured connection. Unfortunately, this solution is not end-to-end secure given the fact that the wireless carrier can "see" the e-mail message before it is sent to the end-user.
- **Symmetric Key encryption security**: Certain devices such as RIM's Blackberries are built-in with symmetric encryption capabilities to access corporate e-mail. In this case, the deployment assumptions are the same as those for the 'Point-to-point security' scenario. However, e-mail communication is secured by encrypting the e-mail at the enterprise side with a shared encryption key, which is also present at the wireless device. Under this solution, the link between the carrier and the e-mail server does not have to be secured, as the payload is encrypted at the application layer. This benefit of this solution is that it is end-to-end private. The limitation of this solution is that it assumes a pre-existing relationship between the enterprise sending the e-mail and the end-user; therefore this solution is best applied to corporate wireless applications.

End-to-end data privacy over e-mail can also be enabled with PKI, which would allow for secure e-mail communication between parties that do not have a pre-existing relationship. E-mail privacy is carried out in the wired world with the use of S\MIME and PGP, a hybrid of symmetric and PKI-based encryption algorithms. Although S\MIME is supported in PC e-mail browsers such as Outlook or Netscape, it is not supported by current Palm e-mail applications and RIM Blackberries.

10.4.2.5 Voice

Voice communication over regular (wired or wireless) phone lines is not end-to-end secure in general. In fact, governments such as that of the United States have taken steps (through laws such as the *Digital Telephony Bill* or the *Digital Wiretap Law*) to facilitate the wiretap of phone communication systems.

Figure 10–8 Voice



Despite these non-technical issues, voice line security can be implemented in the data network, thus discouraging eavesdroppers on a digital network. The following secure solutions are available in the voice channel:

1. **No Transport Security Needed**: As in SMS and e-mail, this deployment scenario depicts both the voice gateway and the application server residing within the same (trusted) domain. Since no data passes through a public digital network such as the Internet, then there is no need to secure the transport communication between the application server and the voice gateway from outsider threats (however, insider threats still remain.) Therefore, communication security relies upon the phone line security itself.
2. **HTTPS-secured connections between Voice Gateway and the Application Server**: In this solution scenario, there is a secure HTTPS connection established between the voice gateway and the application server. This point-to-point security solution makes most sense when the voice gateway and the application server reside in different domains such as when the voice gateway is hosted by a third party. HTTPS is enabled with all major voice gateways (for example, Motorola and VoiceGenie) for SSL-secured connection between the gateway and the application server.

Finally, there are phone devices and third-party mechanisms that claim to provide voice encryption technology that protects communication between two ends of the phone conversation. However, the security of these technologies is not well established and some mechanisms have been breached. In addition, these mechanisms are expensive and not scalable, as they require hardware deployment at the client side.

10.4.3 Non-Repudiation

Non-Repudiation refers to the mechanism where accepted transactions cannot be disclaimed and can be openly verified as valid. For example, in the mobile commerce world, payments cannot be denied. Non-repudiation would allow for such transactions to be openly verifiable and undeniable by the parties involved.

Non-repudiation mechanisms are based on digital signatures, which are analogous to regular ink signatures. Among the many digital signature schemes are DSA (Digital Signature Algorithm), Schnorr's signature and RSA signature - DSA being the most widely used, since the U.S. government has used it as the Digital Signature Standard (FIPS 186). In the wireless world, digital signature mechanisms vary from device to device based on the device's capabilities.

Below are the different means of generating digital signatures across several devices. However, the developer must provide code integration for these non-repudiation mechanisms.

- **WAP**: Digital signature mechanism is carried out through WMLScript's `signtext()` mechanism (available with WAP 2.0.)
- **SMS**: the non-repudiation service sends an SMS to the end user GSM phone requesting a signature. The SMS device detects the signature request and asks the user to enter a PIN (to authenticate the user locally) to start the signing process. The end user, after reviewing the content to be signed, authorizes (or rejects) the signing and the encryption-enabled SIM chip on the device proceeds with the signature.
- **E-mail**: non-repudiation can be enabled for e-mail clients that have signature capabilities such as SMIME enabled clients. This is currently only possible for PC e-mail clients.

11

Mobile Single Sign-On

This chapter, through the following sections, describes Single Sign-On (SSO) for OracleAS Wireless.

11.1 Overview of Mobile Single Sign-On for OracleAS Wireless

Users access the OracleAS Wireless server using mobile or wireless devices, such as personal digital assistants (PDAs) and cellular phones. As in PC-based systems, the authentication mechanism is Oracle Application Server Single Sign-On. All Oracle Application Server 10*g* Release 2 (10.1.2.02) components use SSO for user authentication. The Oracle Internet Directory (OID) is the single point for storing all of the user-related information. The integration of Oracle products with SSO and OID provides:

- Support for partner applications, which take full advantage of the SSO framework, as well external applications for support of legacy and third-party products.
- Seamless integration with Oracle's middle-tier Web portal product, Oracle Portal.
- Management of user information in an external directory.
- Integration with SSO technologies for other, non-Oracle applications.

Users authenticate only once and can access any SSO partner application.

Selecting the *Wireless* option when installing *Oracle Application Server* results in the automatic registration of the OracleAS Wireless and Voice Portal gateway for mobile devices with the SSO server. For more information see the *Oracle Application Server Single Sign-On Administrator's Guide* and also Appendix 16.9, "Re-Registering the OracleAS Wireless Portal Services URL Reference in OracleAS Portal".

11.1.1 Oracle Application Server Wireless Concepts and Architecture

Wireless products communicate with Oracle Application Server using either wireless markup language (WML) or HTML. Cellular phones use WML; PDAs use HTML. Because these devices request URLs using Wireless Access Protocol (WAP) and other non-HTTP protocols, hardware gateways must be used to convert messages to (and from) HTTP.

The heart of OracleAS Wireless is the Wireless and Voice Portal. It serves as a browser for interactions between the wireless device and the SSO server and for interactions between the wireless device and Oracle applications. The Wireless and Voice Portal server performs the following functions:

- It authenticates the user directly to the SSO server.
- It serves private pages of its own.
- It serves as a proxy browser for external, SSO-protected applications by passing requests to these applications, which then perform SSO authentication.
- It converts Oracle Application Server Wireless XML to the appropriate device markup language (either WML or HTML).

In the Wireless and Voice Portal framework, external applications are partner applications that are integrated with the Oracle Application Server SSO Software Development Kit (SDK). The Wireless and Voice Portal treats these applications as public applications even if they are not. A Wireless and Voice Portal instance uses an HTTP adapter to serve as a proxy browser for such applications.

11.2 Wireless Single Sign-On

The mobile user has two SSO authentication options: authenticating directly from the Wireless and Voice Portal home page, or requesting a partner application which then performs the authentication.

This section covers the following topics:

- Authenticating Through the Wireless and Voice Portal
- Authenticating by Requesting a Partner Application

11.2.1 Authenticating Through the Wireless and Voice Portal

The mobile user authenticates from the Wireless and Voice Portal public page either by requesting a private application or by an explicit login request (identified by the URL parameter, `PAlogin=true`) to the SSO server.

Figure 11–1 Interactions Between Oracle Application Server Wireless and the Login Server

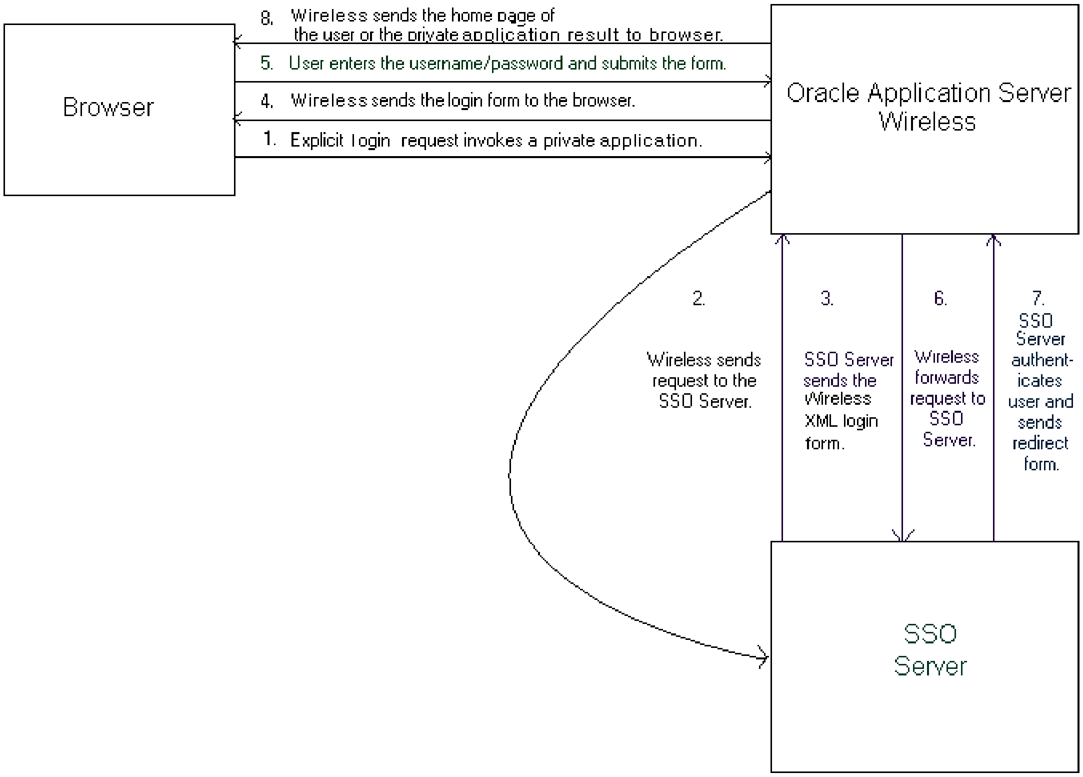

8. Wireless sends the home page of the user or the private application result to browser.
5. User enters the username/password and submits the form.
4. Wireless sends the login form to the browser.
1. Explicit login request invokes a private application.
Browser
Oracle Application Server Wireless
2. Wireless sends request to the SSO Server.
3. SSO Server sends the Wireless XML login form.
6. Wireless forwards request to SSO Server.
7. SSO Server authent-icates user and sends redirect form.
SSO Server

Figure 11–1 depicts the events from the login request to the application result returned to the user as follows:

1. The mobile user accesses the OracleAS Wireless and Voice Portal by entering a URL of the following form:

   ```
   http://<host>:<port>/ptg/rm
   ```

   The *Wireless and Voice Portal* public page appears, displaying links for public and private Wireless and Voice Portal applications.

2. The user requests a private application or selects the key icon that invokes the SSO page. (Figure 11–2 depicts the portion of this page where users enter their names.)

3. The SSO server searches for the encrypted SSO cookie. If the cookie is present, then the server uses it to identify the user. The server then sends the single sign-on redirect form (Step 7). This occurs if the user is already authenticated by an external partner application (Section 11.2.2). If the cookie is not present, then server sends the OracleAS Wireless XML login form to Wireless and Voice Portal.

4. Wireless and Voice Portal transforms the OracleAS Wireless XML login form to the appropriate markup language and sends the converted form to the device browser.

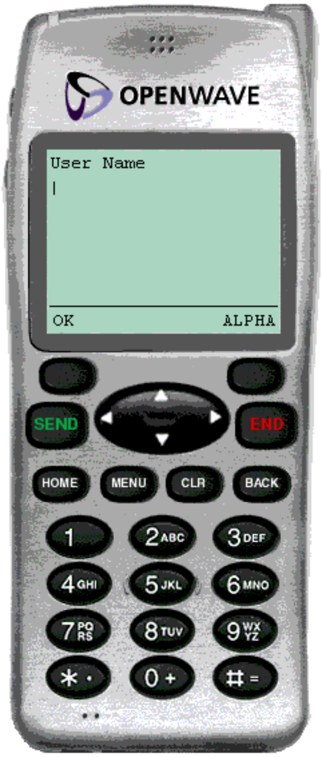
5. The user submits the login form with the user name and password.

6. The Wireless and Voice Portal forwards the login form to the SSO server.

7. The SSO server authenticates the user. If authentication succeeds, then the server sends the Wireless and Voice Portal the SSO redirect form. If the authentication fails, then the SSO server sends a login form (Step 3).

8. The Wireless and Voice Portal sends the user the home page or the requested URL.

Figure 11–2 The Wireless Single Sign-On Page: the User Name Field



User Name

OK ALPHA

11.2.2 Authenticating by Requesting a Partner Application

Using the mobile device, the user may also authenticate to the SSO server by requesting URLs for other partner applications. In this case, the authentication redirection agent is not the Wireless and Voice Portal, but an application integrated with the single sign-on SDK.

The first request to the OracleAS Wireless and Voice Portal (`http://<server>:<port>/ptg/rm`) returns the home page of the anonymous user (a guest user), or the home page of the identified virtual user.

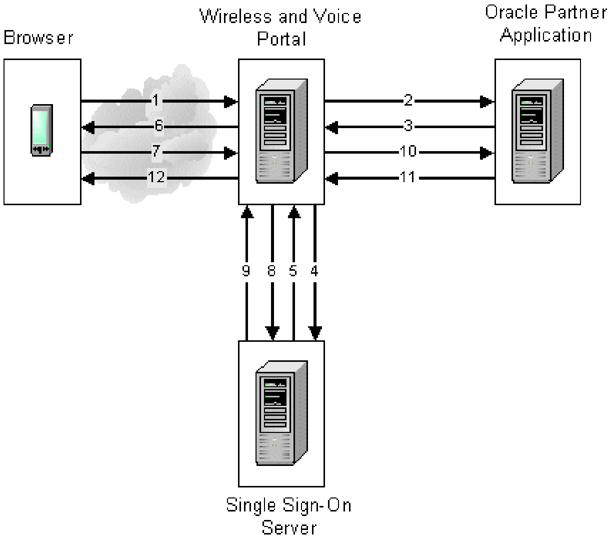
Note: A virtual user is a user who accesses a OracleAS Wireless site, but does not register. When this occurs, OracleAS Wireless detects the user and creates a virtual user account for that user.

An anonymous user is a user who has not registered with OracleAS Wireless but tries the applications as a guest user. The User Manager can create an anonymous user *guest* account for each user group. All of the unregistered users share this account. They cannot, however, personalize applications.

From that point, the user can access public (unsecure) applications or can explicitly log in to the secure applications, which are assigned to that user. The unauthenticated user can execute HTTP Adapter-based public applications, which point to an SSO-based partner application (such as Oracle Portal). The partner application may complete the SSO-based user authentication.

Figure 11–3 illustrates the authentication sequence:

Figure 11–3 Authenticating by Requesting a Partner Application



The authentication sequence (as depicted in Figure 11–3) is as follows:

1. An unauthenticated user requests a partner application.
2. The Wireless and Voice Portal sends the request to the partner application, using an HTTP adapter situated on its back end.
3. If the URL requested is protected, then the partner application issues an HTTP redirect to the SSO server.
4. The Wireless and Voice Portal follows the redirected URL.
5. The SSO server looks for the encrypted SSO cookie, which is set in the Wireless and Voice Portal browser. If the cookie is present, then the server uses it to identify the user. The server then sends the SSO redirect form (Step 9). If the cookie is not present, then the server sends the mobile XML login form to Wireless and Voice Portal.
6. The Wireless and Voice Portal converts the OracleAS Wireless XML login form to the appropriate markup language and delivers the converted form to the device browser.
7. The user submits the login form with the user name and password.
8. The Wireless and Voice Portal passes the login request to the SSO server.
9. Upon successful authentication, the SSO server sends a redirect form that points to the partner application.
10. Wireless and Voice Portal follows the redirect form. At this point, the Wireless and Voice Portal, knowing that authentication has been successful, updates the user's session.
11. The partner application serves content in OracleAS Wireless XML.
12. The Wireless and Voice Portal converts the OracleAS Wireless XML content to the appropriate markup language and delivers the converted content to the device browser.

11.2.3 Authenticating by mod_osso

The OracleAS Wireless Tools, which are used by developers and administrators, as well as those intended for end-users (such as the OracleAS Wireless Customization Portal), authenticate users with mod_osso, which is a module plugged into Oracle HTTP Server. All of the Web-based OracleAS Wireless applications running behind the Oracle HTTP Server are treated as a single partner application. Users can access the applications appropriate to their roles and privileges after single sign-on.

The Wireless and Voice Portal uses the value of the HTTP header `OssoUser_Guid` to identify the mod_osso-authenticated user.

> **Note:** When executing HTTP Adapter-based applications pointing to external partner applications, the mod_sso-authenticated user must be authenticated again, because the SSO cookies are stored in the PC browser for these users.

11.2.4 Authenticating Through Voice

Voice authentication is accomplished by OracleAS Wireless locally using the account number and the PIN of the user.

> **Note:** An authenticated user accessing external SSO partner applications from a voice device must re-authenticate (using username and password).

11.3 Wireless Single Sign-Off

OracleAS Wireless Server participates in SSO global logout for sign off. The following steps detail the interactions between OracleAS Wireless, the SSO Server and Partner Applications.

11.3.1 Logging Out from Oracle Application Server Wireless

The user clicks OracleAS Wireless **Logout** to sign off.

1. The user sends a an OracleAS Wireless Logout request (identified by URL parameter `PAlogoff=true`).
2. The Sign Off implementation of OracleAS Wireless sends an HTTP request to the SSO Sign-Off URL.
3. The SSO server returns the OracleAS Wireless XML global logout page and a special HTTP header (X-Oracle-SSO-logout with value = true). The global logout page contains one image for each partner application that has the user session.
4. OracleAS Wireless sends HTTP requests to each image link. This is done so that the user's session gets cleaned up in all the partner applications.
5. OracleAS Wireless terminates the user's session.
6. If Logout is accomplished through OracleAS Wireless link, then the home page of the guest user is returned.

11.3.2 Logging Out from a Partner Application

The authenticated user clicks the logout link on the page returned by the SSO-based partner application. In this case, the logout link points to the SSO sign-off URL.

1. The user clicks on the logout link which points to the SSO sign-off URL.
2. The SSO server returns the OracleAS Wireless XML global logout page and a special HTTP header (X-Oracle-SSO-logout with value = true). The global logout page includes one image for each partner application which was active in user session.
3. OracleAS Wireless sends HTTP requests to each image link to clean up the user's session in all the partner applications.
4. OracleAS Wireless terminates the user's session.
5. OracleAS Wireless returns the user's home page if the user has logged in through the OracleAS Wireless and Voice portal. OracleAS Wireless returns the done_URL of the global logout page if the user logged in by requesting a partner application.

11.3.3 Logging Out from a Web-Based Oracle Application Server Application

Since all Web-based Oracle Application Server applications are authenticated through mod_osso, and are treated as a single partner application, logout from any application triggers global sign-off . The user cannot access any of the applications until the user signs on through mod_osso again.

11.4 The OracleAS Wireless Change Password Page

The OracleAS Wireless user can view only two SSO pages: the *Login* page and the *Change Password* page. Unlike its PC counterpart, the OracleAS Wireless *Change Password* page appears only when users try to log in to the SSO server with an expired password. OracleAS Wireless users have no access to the *Change Password* link on the *SSO Administration* page.

12 Activity Logging

This chapter describes activity logging for Oracle Application Server Wireless.

12.1 Overview of Activity Logging

The OracleAS Wireless Performance Monitor provides system administrators with information on the running status of Multi-Channel Server, **Notification Engine**, **Messaging Server**, data feed engine, and the **Async Listener**. The Performance Monitor also provides statistical information, enabling system administrators to study past performance and historical data to perform future trend analysis.

OracleAS Wireless integrates with the OEM (Oracle Enterprise Manager) framework to provide a Web-based monitoring tool which displays metrics for diagnosis based on the data logged.

12.1.1 Overview of Activity Logger Internals

The Activity Logger provides the common logging framework used by the runtime components. Database logging is handled asynchronously because the runtime logging on the database carries a huge overhead. The runtime data is generated as files, which are less expensive. The data thus generated is picked up by the Performance Monitor framework and written onto the database. In this way, database logging is handled asynchronously without impacting the runtime performance of the respective servers.

For the Multi-Channel Server, the logging process is handled in the callback of the different events, which are generated (that is, at the beginning of a session and its end). These events are enabled by default for logging purposes. If the administrator chooses not to generate the logging, then there is a provision to turn off the Wireless web server logging. When this happens, the callbacks do not generate log files. For other modules, such as the Notification Engine, Async Listener, and Transport Server, logging into the files occurs when the corresponding request is fulfilled. The Data Feeder logs the runtime data directly to the database in batches.

The generated log files follow a common directory structure, which can be configured using the OracleAS Wireless system management functions at the node (process) level. The top level Logging Directory is specified here, the Logger Framework, which all modules use, creates sub-directories: process, status and archive. At runtime, the log files generated by the different modules have distinct file suffixes. These files are stored in the process directory and the file names and the machine name are enqueued into a `SYS_LOGGER_QUEUE`. The file can be made available for processing based on a configurable file size. Additionally, OracleAS Wireless supports log file aging by which the log file is automatically made available for processing after a fixed time. This ensures that the skew introduced by the asynchronous nature of the logging process is reduced. The log file age (also known as close frequency) can be configured using the system management functions for the site-level configuration of the Performance Monitor.

The modules (which generate these log files with distinct suffixes), provide a Database Log File Handler Class, which processes these files. The handler classes are created by extending a common abstract class, which provides the connection and directory and file information. The handler to suffix mapping is pre-seeded in OracleAS Wireless during installation.

The Performance Monitor starts up multiple threads, each containing an instance of the different handlers. Each logger thread dequeues the filenames belonging to the local machine, inspects the file suffix and delegates it to the corresponding handler class for further processing.

The administrator can control the number of Performance Logger threads using the system management functions for the process-level configuration.

12.1.2 Activity Log Table Description

> **Note:** Since these tables tend to grow during the life of the servers, you should purge the data from these tables periodically.

PTG_SERVICE_LOG

Table 12–1 describes the PTG_SERVICE Log.

Table 12–1 Service Activity Log

Column Name	Description
Service_id	The Object Identifier for the invoked service (application).
Service_name	The name of the invoked service.
ptg_instance_id	The unique identifier identifying the instance.
final_service_id	The Object Identifier of the final service (that is, master service folder).
final_service_name	The name of the final service.
session_id	The Session Identifier of the Session in whose context the service is invoked.
bookmark	The application bookmark.
service_type	The type of service.
invocation_hour	The hour when the service was invoked.
invocation_time	The date when the service was invoked.
response_time	The response time for the service.
request_status	The status of the request. Non-zero values indicate the error number.
error_description	The error message (if there was an error while invoking the service).
user_id	The Object Identifier for the user.
user_name	The name of the user.
remote_address	Gateway IP address and host name.

Table 12–1 (Cont.) Service Activity Log

Column Name	Description
host_id	Host IP address and name.
logical_device	The device where the application was invoked.
external_user_id	The external user id of the which forwarded this request.
external_user_name	The external user name of the which forwarded this request.
adapter_type	The type of the adapter which is servicing this request (not logged currently).
adaptor_time	Time taken by the adapter to service this request.
transformation_time	Time taken by the transformer to service this request.
timestamp	Logged event timestamp (generated by trigger).

Table 12–2 describes the DATAFEEDER_METRICS activity log.

Table 12–2 DATAFEEDER_METRICS

Column Name	Description
HOST_NAME	The host name of this data feeder.
INSTANCE_NAME	The instance name of this data feeder.
FEED_NAME	The name of this data feeder.
UPDATE_DATE	The date and time of this batch run.
ACTUAL_BATCHTIME	The actual time spent on this batch.
DOWNLOADED_ROWS	The publishing rate (data rows stored).
ERROR_DESCRIPTION	Errors encountered for this batch for future use.

Table 12–3 describes the PTG_ALERT_ENGINE_STATS log.

PTG_ALERT_ENGINE_STATS

Table 12–3 Notification Engine Activity Log

Column Name	Description
host_name	The host name of the machine this alert server instance is running on.
instance_name	The alert instance name.
malert_name	The name of the master alert service which generates this alert message.
malert_oid	The Object Identifier of the master alert service which generates this alert message.
subscriber_name	The name of the subscriber to receive this alert message.
device_address	The device address this alert message is delivered to.
device_oid	The device address object identifier.
device_type	The type of the device.
message_id	The message id generated by the message gateway for this alert message.

Table 12–3 (Cont.) Notification Engine Activity Log

Column Name	Description
message_length	The length of this alert message.
message_status	The dispatch status of this alert message.
dispatch_time	The time stamp of this alert message being dispatched to the message gateway.
error_description	The error message - if there was an error while dispatching this alert message.

ASYNC_STATISTICS_LOG

Table 12–4 Async Listener Activity Log

Column Name	Description
host	Name of the host where the Async server is running.
instance_id	The unique id to identify an instance of the Async server.
source_addr	The source address of the received message.
dest_addr	The destination address of the received message.
delivery_type	The network delivery type of the message. The possible values are: ■ WAP-Push ■ SMS ■ Voice ■ e-mail ■ Fax ■ Two-Way Pager ■ One-way Pager
encoding	The character encoding for the message.
queue_size	The number of messages waiting in the queue when the message is received.
msg_rcv_time	The message received time.
msg_rcv_hour	The message received hour.
start_execute_time	The time to start invoking the service requested from the message.
end_execute_time	The time to finish the service invocation.
error_description	The error description on failure of the service invocation.
service_id	The ID of the service the user is requesting to access.
async_name	The Async short name of the service the user is requesting to access.
message_size	The size of the message.
timestamp	Time when the message is logged into the database

TRANS_LOG

Table 12–5 Message Server Activity Log

Column Name	Description
MESSAGE_ID	The message id assigned by the transport, which is unique for every message.
MESSAGE_TYPE	The type of the message, which can be 'R' for received message, 'S' for message to send.
DELIVERY_TYPE	The delivery type, which can be: ■ WAP-Push ■ SMS ■ Voice ■ e-mail ■ Fax ■ Two-Way Pager ■ One-Way Pager
REQUEST_INSTANCE_HOST	The transport instance host on which the message is accepted. For a sending message, this is the host of the client; for a received message, this is the host of the driver.
REQUEST_INSTANCE_ID	The OracleAS Wireless instance id on which the message is accepted. For a sending message, this is the instance id of the client. For a received message, this is the host of the transport server that the driver is on.
REQUEST_BEGIN_TIME	The time the message is to be accepted. For a sending message, it is the time the send method is called. For a received message, it is the time the `onMessage` method is called. All time is Java system time.
REQUEST_END_TIME	The time the message is accepted. For a sending message, it is the time the send method returned. For a received message, it is the time the `onMessage` method returned.
HANDLE_INSTANCE_HOST	The host name on which the message is dequeued to a process. For a sending message, it is the host on which the driver ran. For received message, it is the host on which the driver ran.
HANDLE_INSTANCE_ID	The OracleAS Wireless instance id on which the message is dequeued to process.
HANDLE_BEGIN_TIME	The time the dequeue method to be called.
HANDLE_END_TIME	The time the message is processed. For sending message, the message is sent. For received message, the message is processed by the listener.
ENQUEUE_BEGIN_TIME	The time the enqueue call started.
ENQUEUE_END_TIME	The time enqueue call returned.
DEQUEUE_BEGIN_TIME	The time the dequeue call started.
DEQUEUE_END_TIME	The time the dequeue call returned.
PROCESS_STATUS_CODE	The status code of the message processing, which can have the values *unknown*, *failed*, *succeeded*, *ignored*.
PROCESS_BEGIN_TIME	The time the processing call was called. For a sending message, the driver's send method was called. For a received message, the listener's `onMessage` method was called.

Table 12–5 (Cont.) Message Server Activity Log

Column Name	Description
PROCESS_END_TIME	The time the processing call returned. For a sending message, the driver's send method returned. For a received message, the listener's `onMessage` method returned.

System Logging

The System Logger logs the runtime debug log information generated by the runtime processes. The OracleAS Wireless Server generates log information, which is stored in the log file. The different levels of logging and the log file size can be configured as follows:

To configure the System Log file using OracleAS Wireless system management at either the site or process level:

- Enter a name for the log file name pattern. The default is *sys_panama.log*.

  This pattern enables you to identify the log file generated by the different server processes. Currently, the only supported pattern is `<filename>{0}.log`. For example, sys_panama{0}.log would generate a file with a name sys_panama<timestamp in long>.log. Using this pattern enables administrators to identify log files pertaining to the different server processes based on their start timestamp. The setting of the pattern is optional.

  At the OracleAS Wireless server or host level, the log directory may be specified using Wireless Management. The default log directory is the default temp directory for that operating system (typically `c:\temp` for windows and `/var/tmp` on UNIX).

- In the *Maximum Log File Size* field, enter the maximum number of log file size (in bytes).
- Select a log level. The log can contain any of the following: *Warning*, *Error*, or *Notify*. The default is *Warning*, *Error*, and *Notify*.

See also Section 3.4.

13

Optimizing Oracle Application Server Wireless

This chapter, through the following sections, discusses factors that enable application developers to optimize Oracle Application Server Wireless.

13.1 Overview of OracleAS Wireless Optimization

Oracle Application Server Wireless, when installed, initializes a default setup that is appropriate for the performance of most applications. However, you may need to use additional tuning knobs to adjust performance, since applications vary in features, hardware setup, and performance requirements.

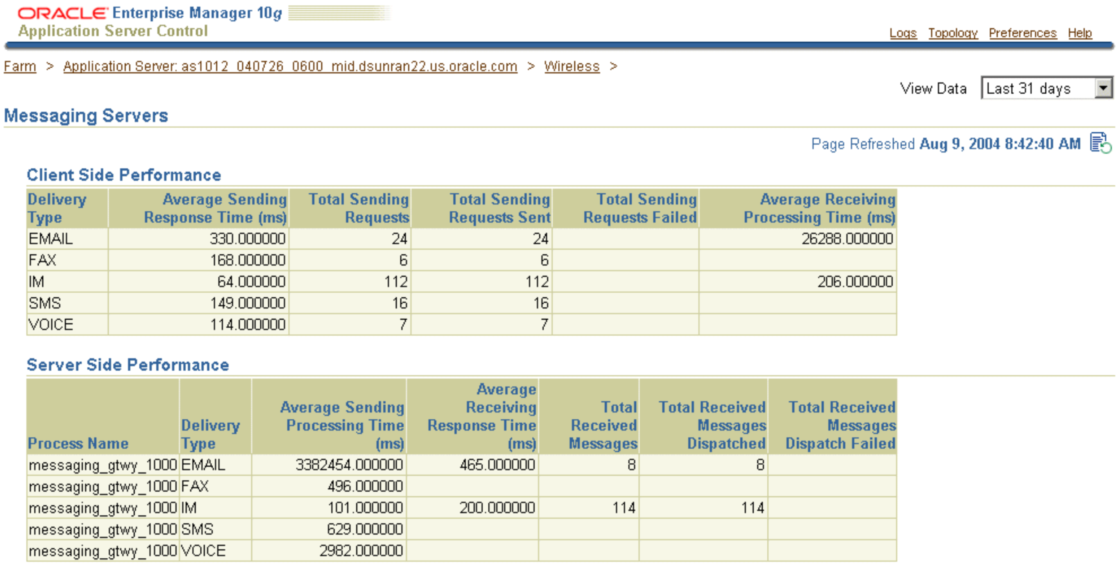
This chapter discusses the tuning options and methods available within Oracle Application Server Wireless and the performance logger utility. It also discusses JVM tuning, JDBC connection performance, and TCP/IP stack tuning.

Note: Throughout the documentation, you can substitute UNIX for Solaris in all instances except for this chapter. The tuning knobs described in this chapter are Solaris-specific.

13.2 Transport Performance Monitoring

You can view the performance statistics of the Transport system from the Oracle Enterprise Manager 10*g* Application Server Control by first selecting the middle-tier node on the *Farm* page of the Oracle Enterprise Manager 10*g* Application Server Control and then by clicking **Wireless** on the *Application Server Home* page to access the OracleAS Wireless system management functions described in Chapter 3, "Managing the OracleAS Wireless Server". Click the *Site Performance* tab of the *Wireless Home* page. From the *Component Performance* Section, select **Messaging Servers**. The **Messaging Server** performance page appears (Figure 13–1).

Figure 13–1 The Messaging Server Performance Metrics



Farm > Application Server: as1012_040726_0600_mid.dsunran22.us.oracle.com > Wireless >

View Data: Last 31 days

Messaging Servers

Page Refreshed **Aug 9, 2004 8:42:40 AM**

Client Side Performance

Delivery Type	Average Sending Response Time (ms)	Total Sending Requests	Total Sending Requests Sent	Total Sending Requests Failed	Average Receiving Processing Time (ms)
EMAIL	330.000000	24	24		26288.000000
FAX	168.000000	6	6		
IM	64.000000	112	112		206.000000
SMS	149.000000	16	16		
VOICE	114.000000	7	7		

Server Side Performance

Process Name	Delivery Type	Average Sending Processing Time (ms)	Average Receiving Response Time (ms)	Total Received Messages	Total Received Messages Dispatched	Total Received Messages Dispatch Failed
messaging_gtwy_1000	EMAIL	3382454.000000	465.000000	8	8	
messaging_gtwy_1000	FAX	496.000000				
messaging_gtwy_1000	IM	101.000000	200.000000	114	114	
messaging_gtwy_1000	SMS	629.000000				
messaging_gtwy_1000	VOICE	2982.000000				

This page displays the client side and server side Messaging Server performance metrics. For each of the Messaging Server performance metrics, Wireless displays performance data by process name and delivery type (for example, SMS).

The client side performance metrics include:

Average Sending Response Time

The average time of a sending method. On the client side, a sending method is called to send a message. This time is the period from when the method is called to the time the method returns. When the method returns, the message is saved in a database persistently, but is not delivered.

Total Number of Sending Requests

The total time that the sending method is called by the client process. A sending method called once to send a message to a set of destinations counts as a single sending request.

Total Number of Sending Requests Sent

The total number of successful calls, where a message is delivered to a proper gateway and its receipt is acknowledged. The client process can call the sending method many times to send many messages. Some of these requests fail, as in the case where a destination cannot be reached. Other requests could be undergoing processing.

Total Number of Sending Requests Failed

The total number of all calls that are known to have failed.

Average Receiving Process Time

The performance of the listener in terms of the time taken by the `onMessage` call-back.

The server-side performance metrics include:

Average Sending Process Time

The performance of a driver in terms of the time taken by the sending method of the driver. The driver performance is measured by delivery type (for example, SMS), process time (the time taken by a driver to send a message to the proper gateway), dequeue time, and driver process time. When you measure the performance of the transport system, you can deduct the process time, because the transport system is waiting while the driver sends a message. If the driver is fast, then the system does not wait long.

Average Receiving Response Time

Once a transport driver receives a message, the message is passed to the transport system by an `onMessage` method. The response time is the time taken by the `onMessage` method. Once the `onMessage` returns, the received message is saved in a database for dispatching.

Total Number of Received Messages

The total number of times the transport drivers call the `onMessage` call-back method.

Total Number of Received Messages Dispatched

The total number of received messages which are dispatched to, and are accepted by, the listeners. Among received messages, some may be in processing. Others may not have been dispatched to listeners, or listeners may have failed to process dispatched messages.

Total Number of Received Messages Dispatch Failed

The total number of received messages which failed to dispatch to a listener.

For more information on the *Site Performance* tab, see Section 3.8.

13.2.1 Factors Affecting Transport Performance

This section describes the factors that affect the messaging server performance. Topics include:

- Section 13.2.1.1, "The Sending and Receiving Threads of a Driver"
- Section 13.2.1.2, "Messaging Server Client Threads"
- Section 13.2.1.3, "JDBC Connection Pool"
- Section 13.2.1.4, "Performance Monitor Process"
- Section 13.2.1.5, "AQ Tuning"
- Section 13.2.1.6, "Cleansing Messaging Server Tables"

13.2.1.1 The Sending and Receiving Threads of a Driver

The sending and receiving threads are the number of threads spawned by the Messaging Server to call a driver's send and receive methods.

Every delivery type has an associated driver queue in which the Messaging Server queues the messages that are to be sent using a specific delivery type. Every messaging client has an associated service queue in which the Messaging Server queues the messages that it receives from the gateway.

By increasing the number of sending threads, you also increase the number of threads that both dequeue the message from the driver queue and pass it to the send method that performs the send operation by submitting the message to the gateway. If the dequeue rate from the driver queue is low, then increasing the number of sending threads enhances performance.

Likewise, by increasing the number of receiving threads, you also increase the number of threads that both receive messages from the gateway and queue them to the service queue for a messaging client. The messaging client then receives messages from this queue. Increasing the number of working threads improves performance if the enqueue rate is low or if there are many pending messages in the gateway.

Tips:

- To find the optimum number of sending threads, study the impact of varying the number of sending threads on the resource utilization of the database machine (mainly I/O and CPU) and the dequeue rate from the driver queue
- To find the optimum number of receiving threads, study the impact of varying the number of receiving threads on the resource utilization of the database machine (mainly I/O and CPU) and the enqueue rate of the service queue.
- Setting the number of sending threads for the driver between 7 and 10 and the number of receiving threads between 3 and 5 at average load yields decent performance from the Messaging Server.

.

13.2.1.1.1 Finding the Queue Table for a Driver's send Method

The name of the table that queues the messages for the driver to send is of the form `trans_t_<queue_id>`. The `<queue id>` for a delivery type is found in the trans_driver_queue table. For example, if the `<queue id>` for the SMS delivery type is 1, then all of the messages sent using the SMS delivery type are in the table, `trans_t_1`.

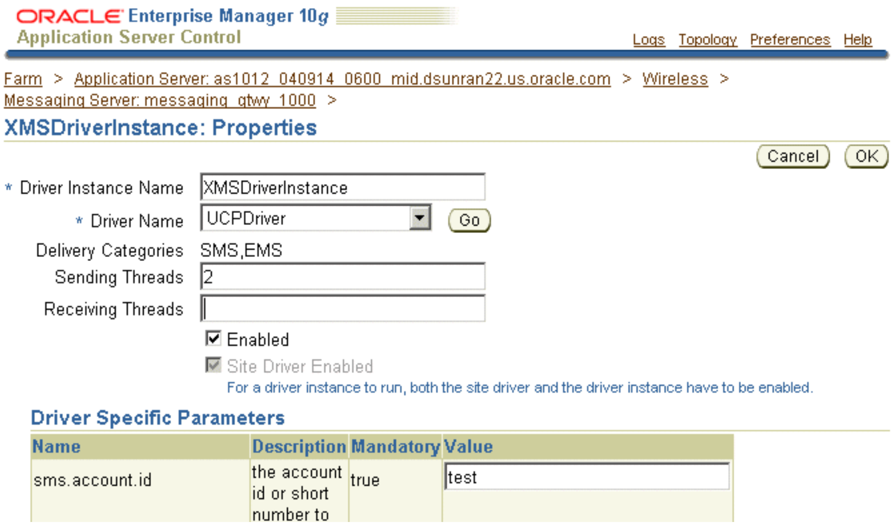
13.2.1.1.2 Finding the Queue Table for a Service or Client

The messaging server receives a message from the gateway and then queues it into AQ if any client or service has been registered for the message. All messages received by a driver for a particular service or client is queued in a the queue table of the form trans`_t_<queue_id>`. The `<queue_ id>` for a service name is found in the trans_service_queue table. For example, if the `<queue_ id>` for a service called *ASYNCAGENT* is 2, then all of the messages for this service are stored in the queue table called `trans_t_2`.

13.2.1.1.3 Increasing the Number of Sending and Receiving Threads for Driver Instance

You can increase the number of sending and receiving threads by editing a driver instance. To edit a driver instance, first select a Messaging Server process from the *Standalone Processes* section of the OracleAS Wireless System Manager *Home* page (accessed through Oracle Enterprise Manager 10g Application Server Control as described in Section 13.2) which has the driver instance with the thread values that you need to edit. From the *Driver Instances* section of the detail page for the selected Messaging Server Process, select the driver and then click **Edit**. The *Properties* page for the selected driver appears (Figure 13–2), with its fields populated by the values set for the selected driver. Change the values for the *Sending Threads* and *Receiving Threads* parameters as needed and then click **OK** to commit any changes.

Figure 13–2 The Driver Instance Properties Page



Farm > Application Server: as1012_040914_0600_mid.dsunran22.us.oracle.com > Wireless > Messaging Server: messaging_gtwy_1000 >

XMSDriverInstance: Properties

* Driver Instance Name	XMSDriverInstance
* Driver Name	UCPDriver
Delivery Categories	SMS,EMS
Sending Threads	2
Receiving Threads	

☑ Enabled

☑ Site Driver Enabled

For a driver instance to run, both the site driver and the driver instance have to be enabled.

Driver Specific Parameters

Name	Description	Mandatory	Value
sms.account.id	the account id or short number to	true	test

13.2.1.2 Messaging Server Client Threads

The Messaging Server client is a service that is registered with the Messaging Server for sending and receiving messages (for example, the Async Listener and the Notification Engine are clients of the Messaging Server). A service queue is associated with every client that is registered to receive messages from the Messaging Server. The service queue contains all of the messages intended for the client and received by the Messaging Server.

Messaging Server client threads are the number of threads that the client uses to dequeue the messages from its service queue. To find the name of the service queue table for a client, refer to Section 13.2.1.1.2. If the dequeuing rate from the service queue is low, then you can improve performance by increasing the number of Messaging Server client threads. You can increase the client threads for Messaging Server clients using the appropriate configuration page for the Messaging Server client in the OracleAS Wireless system management described in Chapter 3, "Managing the OracleAS Wireless Server". For example, increasing the thread pool size of the Messaging Server client for the Async Listener configures the Messaging Server client threads for the Async Listener. For more information, see Section 3.10.2.1.

13.2.1.3 JDBC Connection Pool

The JDBC Connection Pool parameter defines the size of the database connection pool. If you increase the number of driver or client threads to more than 10, then increasing the maximum number of connections from 10 (the default value) to 50 ensures decent performance at peak load.

13.2.1.4 Performance Monitor Process

If a very high load is expected to hit the Messaging Server and if performance data logged by the Performance Monitor is not required, then stopping the Performance Monitor process will improve performance. At high loads, this process has been observed to consume considerable resources, resulting in low throughput.

13.2.1.5 AQ Tuning

AQ (Advanced Queuing) operations result in high number of insertions and deletions from the database. Hence, I/O values on the database will be high and will need careful tuning. Based on the volume of operations, you may want to increase the number of I/O controllers on the machine.

In the test environment, the following observations have been verified.

- With the 3 I/O controller, a throughput of 40 messages per second with 7 sending threads was achieved.
- With the 12 I/O controller, a throughput of 100 messages per second with 9 sending threads was achieved.

13.2.1.6 Cleansing Messaging Server Tables

Cleaning database tables can improve performance if the Messaging Server or the Performance Monitor process has been running for a long time. You can delete the following tables if the Messaging Server has been running for a long time and if the listed tables contain many rows:

- Trans_message
- Trans_store
- Trans_ids

Note: Before deleting these tables, be sure that you do not need message details about the messages received and sent by the Messaging Server.

If the Performance Monitor process has been running for a long time and if the listed tables contain many rows, the you can truncate the following tables to improve performance:

- `trans_request_log`
- `trans_handle_log`
- `trans_process_log`
- `trans_enqueue_log`
- `trans_dequeue_log`

Note: Before truncating these tables, be sure that you do not need any performance data.

13.2.1.6.1 Recreating the Transport Repository To recreate the transport repository from scratch, run the following scripts from SQL*Plus connected to the repository as *wireless* user. Be sure to stop the middle tier before running the following scripts, which are located at `wireless/repository/sql` directory:

- `trans_clean.sql`
- `trans_setup.sql`
- `trans_setup.pls`
- `trans_setup.plb`

13.3 Optimizing the Async Listener Performance

The performance logging framework at the Web Server level collects performance-related data for an **Async Listener**. To view this data, you first select an Async Listener process (located in the *Web-Based Processes* section of the OracleAS Wireless System Manager Home page). The detail page for the selected process appears. Clicking the *Performance* tab of the detail page invokes the process's *Performance* page. The page includes the following performance metrics:

Number of Messages Received

The number of messages received, grouped by process ID.

Average Message Response Time (seconds)

The average time a message stayed on the server.

Average Message Queue Size

The average size of the message queue on an hourly basis for today.

Service Access Count

The number of times that each application was accessed today.

User Access Count

The number of messages issued by each user device.

Number of Errors

The number of errors on an hourly basis.

13.3.1 Tuning the Performance of the Async Listener

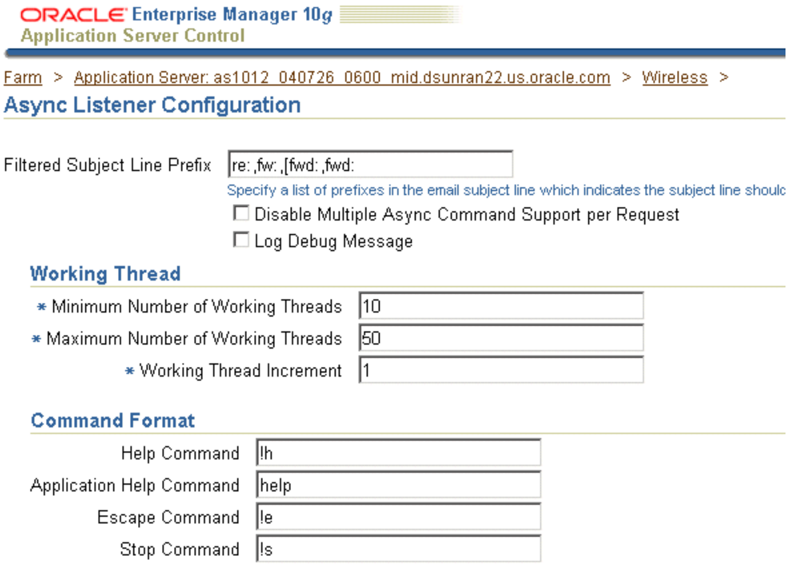
The following sections describe the tuning knobs available in OracleAS Wireless that affect Async Listener performance:

- Section 13.3.1.1, "Tuning the Working Threads for the Async Listener"
- Section 13.3.1.2, "Adjusting the Thread Pool Size of Messaging Server Client"
- Section 13.3.1.3, "Adjusting the Sending and Receiving Threads"

13.3.1.1 Tuning the Working Threads for the Async Listener

The *Async Listener Configuration* page (Figure 13–3) enables you to change the number of working threads for the Async Listener. By default, the value for the *Working Threads* parameter is 10. You can increase this parameter to a higher value to accommodate a higher request rate.

Figure 13–3 Configuring the Async Worker Threads



ORACLE Enterprise Manager 10*g*
Application Server Control

Farm > Application Server: as1012_040726_0600_mid.dsunran22.us.oracle.com > Wireless >

Async Listener Configuration

Filtered Subject Line Prefix	re: ,fw: ,[fwd: ,fwd:
	Specify a list of prefixes in the email subject line which indicates the subject line shoulc
	□ Disable Multiple Async Command Support per Request
	□ Log Debug Message

Working Thread

* Minimum Number of Working Threads	10
* Maximum Number of Working Threads	50
* Working Thread Increment	1

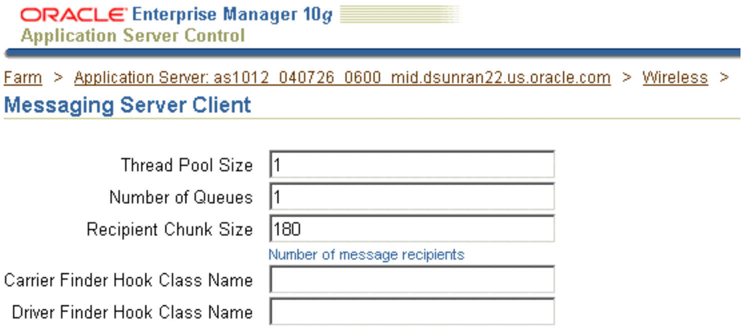
Command Format

Help Command	!h
Application Help Command	help
Escape Command	!e
Stop Command	!s

13.3.1.2 Adjusting the Thread Pool Size of Messaging Server Client

Increasing the size of the thread pool enables the Messaging Server client to handle higher loads. You can adjust the size of the thread pool from the *Messaging Server Client* page (Figure 13–4). To access this page (Figure 13–4), select **Messaging Server Client** (located under *Notification Engine* in the *Component Configuration* section of the *Administration* tab. For more information on configuring the Messenger Sever client, see Section 3.10.2.1.

Figure 13–4 Adjusting the Thread Pool Size



ORACLE Enterprise Manager 10*g*
Application Server Control

Farm > Application Server: as1012_040726_0600_mid.dsunran22.us.oracle.com > Wireless >

Messaging Server Client

Thread Pool Size	1
Number of Queues	1
Recipient Chunk Size	180
	Number of message recipients
Carrier Finder Hook Class Name	
Driver Finder Hook Class Name	

13.3.1.3 Adjusting the Sending and Receiving Threads

You can also speed up de-queuing and enqueuing by increasing the number of sending and receiving threads. For more information, see Section 13.2.1.1.3.

13.4 Optimizing Data Feeder Performance

Parsing input is a costly operation. The performance of such operations depends largely on the amount of memory available to the Java Virtual machine (JVM). To handle a high feed size, you can increase the heap size of the Data Feeder process. Normally, parsing XML feeds consume larger resources than CSV (comma-separated variable) feeds.

In the test environment, the following observations have been verified.

- With a large XML feed of 25 MB, a throughput of 43 data rows per-second was achieved by using a heap size of 512 MB.
- For a CSV feed of the same volume, a throughput of 48 data rows per-second was achieved.

13.5 Optimizing the Performance of the Oracle HTTP Server

The following sections describe the configuration directives that you can tune in *httpd.conf(located in the ORACLE_HOME/Apache/Apache/conf/ directory)* to enhance performance of the Oracle HTTP Server (OHS).

- Section 13.5.1, "MaxClients"
- Section 13.5.2, "MaxRequestsPerChild"
- Section 13.5.3, "MaxSpareServers"
- Section 13.5.4, "MinSpareServers"
- Section 13.5.5, "Start Servers"
- Section 13.5.6, "Timeout"

13.5.1 MaxClients

This is the maximum number of servers that can run. An optimum number should be used based on load. A low number causes clients to be locked out; a high number of servers consumes more resources.

13.5.2 MaxRequestsPerChild

The number of requests that a child process handles before it expires and gets re-spawned. The default value *0* means that it will never expire. As as result, you should limit this value. Generally, *10000* is sufficient.

13.5.3 MaxSpareServers

This is the maximum number of pre-spawned processes that are available in the pool of the Apache process that handles connections. The suggested value may vary, as *10* will suffice for most requirements.

13.5.4 MinSpareServers

This is the minimum number of child processes that need to be pre-spawned all the time. The value *5* will suffice for most requirements.

13.5.5 Start Servers

The number of servers to start initially. If a sudden load is expected on startup, then this value should be increased.

13.5.6 Timeout

The number of seconds before incoming receives, and outgoing sends the time out. The recommended value is *300* seconds.

13.6 Optimizing the Oracle Process Management and Notification Service (OPMN)

Because the default file descriptor number per JVM is low, you should increase this number to a higher value in the `ORACLE_HOME/opmn/bin/opmnctl` script by modifying (or adding) the following line:

```
> ulimit -n 2048
```

Note: You must bounce OMPN after you increase the value of the default file descriptor.

13.7 Optimizing the Database Connections

Oracle Application Server uses database connections for Single Sign On (SSO), Oracle Internet Directory (OID), and other connections. Because the default number of connections may not suffice for a high number of users, you should therefore increase this number as users increase. You can increase this number by modifying the relevant files in the database.

13.8 Optimizing the Capacity of Webcache

The Webcache capacity should be set to a high value depending upon the load. For example, if you are hitting 50 requests per second, then you must set the capacity to *1000*. You can increase the Webcache capacity by editing the properties of the Webcache's origin server. To edit the origin server:

1. Click **Web Cache** in the *Application Server Home* page of the Enterprise Manager. The *Web Cache Home* page appears.
2. Click the *Administration* tab.
3. Click **Origin Servers** (located in the *Application* section under *Properties*). The *Origin Server* page appears.
4. Select an origin server and then click **Edit**. The *Edit Origin Server* page appears.
5. Increase the value in the *Capacity* field.
6. Click **OK** to commit the change.

Note: Depending on the incoming requests and the size of documents to be cached, you must also change the maximum incoming connections and the maximum cache size. You can change these values using the *Resource Limits and Timeouts* page (accessed by clicking **Resource Limits and Timeouts** in the *Web Cache* section of the Web Cache *Administration* tab).

13.9 Optimizing JVM Performance

Because Java applications run within the context of the JVM, some default properties of the JVM must be modified to enable the applications to run faster and consume fewer resources.

Since garbage collection (GC) was not a parallel process until Java 1.3.1, it can become a principal performance bottleneck as the number of CPU's increase. Java 1.4.2

implements the concept of generational garbage collections. Generations are memory pools that hold objects of different ages. There are two GC cycles: Minor Collection and Major collection.

Minor Collection

Typically young objects, which comprise the *young* generation, die fast. Minor collection occurs when memory pool (or generation) of young objects fills up. During Minor Collection, live objects from the *young* generation memory pool are eventually copied to a *tenured* pool. The *young* generation consists of Eden, where objects are initially allocated and two survivor spaces. One of these survivor spaces remains empty and serves as the destination where live objects in Eden and the other survivor space are copied. Objects are copied between the survivor spaces until they become old enough to be tenured (copied to the tenured generation).

Major Collection

The collection of older generation, or tenured objects. Typically, a Major Collection is slower than a Minor Collection because it involves all live objects.

The first step in tuning is to observe the frequency of GC by using the following command-line option:

```
> java -verbose: gc classname
```

This command results in output similar Example 13–1.

Example 13–1 GC Frequency

```
> [GC 866K->764K(1984K), 0.0037943 secs]
> [GC 1796K->1568K(2112K), 0.0068823 secs]
> [Full GC 2080K->1846K(3136K), 0.0461094 secs]
> [GC 2047K->1955K(3136K), 0.0157263 secs]
```

For more information on parallel GC, see Section 13.10.1.

13.9.1 Modifying the Default GC Behavior

The following knobs (available within Java 1.4.2) change the default GC behavior by modifying the heap and generation size.

-Xms and -Xmx

The total size of the heap is bounded by the `-Xms` and `-Xmx` values. `-Xms` is the minimum size of the heap and `-Xmx` is the maximum size to which the heap can grow. Having a larger heap reduces the frequency of collections. Increase the heap size as the number of processors increase, since allocation can be done in parallel.

-XXNewSize and -XX:MaxNewSize

These options are specific to Sun Microsystem’s HotSpot VM. The *young* generation size is bounded by these values. A smaller generation, which means a faster rate of Minor Collections and lower frequency of Major Collections, is best suited for Web applications.

By changing these parameters, you can change the frequency of collections as desired by the application.

13.9.1.1 Improving GC Performance

Other knobs that improve GC performance include:

- XX: New Ratio

NewRatio controls the *young* generation size. For example, setting `-XX:NewRatio=3` means that the ration between the young and tenured generation is 1:3.

-XX:SurvivorRatio

Use the parameter `SurvivorRatio` to tune the size of the survivor spaces. For example, `-XX:SurivorRatio=6` sets the ratio between each survivor space and Eden at 1:6. In other words, each survivor space is one-eighth of the *young* generation (not one-seventh, because there are two survivor spaces).

-XX: +UseParallelGC

The throughput collector is a generational collector similar to the default collector but with multiple threads that perform minor collection. Enable the throughput collector using the command-line flag `-XX:+UseParallelGC`.

-XX:ParallelGCThreads

Use the `ParallelGCThreads` command-line option to control the number of garbage collector threads.

-Xss

This is the size of the stack on a per-thread basis. Its default value changes from platform to platform. If the number of threads running in the application is high, then you can decrease the default size. If, for example, the threads require a high stack space for parsing operations and recursive calls, then increasing the stack size can provide significant performance increase.

Tune the value of these options according to the application type. Table 13–1 describes a typical setup for the E420/Solaris box with four 450Mhz processors and four GB RAM to support 2000 concurrent users.

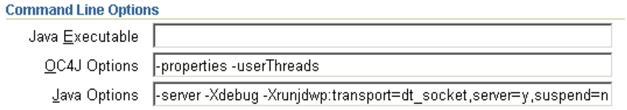
Table 13–1 Typical Setup for the E420/Solaris Box

Options	Recommended Value
`-Xms`	1024m
`-Xmx`	1536m
`-XX: NewRatio`	2
`-XX: SurvivorRatio`	16
`-XX: ParallelGCThreads` (Use with `-XX:+UseParallelGC`)	4
`-Xss`	256K
`-XX:UseLWPSynchronization`	This thread model should be used.

13.9.2 Modifying JVM Parameters

Modify these Java attributes in the *Java Options* field, located in the *Command Line Options* section of the Oracle Enterprise Manger's *Server Properties* page (Figure 13–5). To access this page, first select the **OC4J_Wireless** process on the *Home* page and then select **Administration**. Next, select **Server Properties** (located under *Instance Properties*). The *Server Properties* page appears.

Figure 13–5 Editing the Java Options



Command Line Options

Java Executable	
OC4J Options	-properties -userThreads
Java Options	-server -Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n

13.10 Optimizing the OC4J_Wireless Server Instance

The JVM heap-tuning options `-Xms512m` and `-Xmx1024m` increase the maximum memory size to 1 GB (or more) and enable the OC4J_Wireless server instance to support a large number of concurrent users.

To support higher hit rates, increase the `MaxClients` parameter in `httpd.conf`. For example, setting the `MaxClients` parameter to 1024 in `httpd.conf` allows up to 1024 concurrent HTTP requests. Consequently, increased requests result in an increased number of Oracle Application Server threads in the OC4J_Wireless server instance. To support large numbers of Oracle Application Server threads in the OC4J_Wireless server instance, reduce the thread stack size to 256k using the JVM option, `-Xss256k`. The default thread stack size in the Solaris environment is 512k.

For more information on the JVM options, see Section 13.9.

13.10.1 Enabling Parallel Garbage Collection

For OC4J_Wireless server instances running on multi-CPU machines, set the JVM options to enable the Parallel Garbage Collection (GC) algorithm in JDK 1.4. Set the `ParallelGCThreads` parameter to the number of CPUs in the host. The following JVM options for JDK 1.4 increase the performance of the OC4J_Wireless Server instance in 4-CPU Solaris machines:

```
-XX:+UseParallelGC -XX:ParallelGCThreads=4
```

The following GC tuning parameters provide improved performance for the OC4J_Wireless server instance:

```
-XX:NewRatio=2 -XX:SurvivorRatio=16
```

13.11 Tuning the Performance of the Operating System

This section describes tuning methods for the operating system’s performance of Oracle Application Server Wireless.

13.11.1 TCP/IP Tuning

Correctly tuned TCP/IP settings improve performance. The indicators for changing default parameters are primarily TCP connection drops while making the three-way handshake, and the system refusing connections at a certain load.

Note: The information in this section applies only to Solaris.

Use the following UNIX command to check for TCP connection drops:

```
netstat - s | grep Drop
```

Note the following value:

`tcpListenDrop`, `tcpListenDropQ0`, `tcpHalfOpenDrop`

Any value other than zero suggests the need for changing the tcp connection queue size. While any value for `tcpListernDrop` suggests a bottleneck in executing the `accept()` call and value for `tcpListenDropQ0`, it is an indication of SYN flood or denial-of-services attack.

Use the following UNIX command to check if connections should be replenished more quickly:

```
netstat | grep TIME_WAIT | wc - l
```

Note the number of connections in the `TIME_WAIT` state. If the rate of establishing connections (load) is known, then you can compute the time taken to run out of connections. To ensure that new connections are readily available, you can decrease the `tcp_time_wait_interval` to a low value of 10000 ms.

You can set most of these values using UNIX command ndd. For example:

```
> ndd - set  /dev/tcp tcp_time_wait_interval 10000
```

These parameters (described in Table 13–2), take effect after the application is restarted. They should be added to the system startup file so that they are not lost after a reboot

You must change the `tcp_conn_hash_size` in the file */etc/system* after a reboot.

Table 13–2 Operating System Performance Parameters

Parameter	Setting	Comments
`tcp_time_wait_interval`	10000	The time out for disposing closed connection information. This makes new connections readily available.
`tcp_xmit_hiwat`	65536	The size of the TCP transfer windows for sending and receiving data determines how much data can be sent without waiting for an acknowledgment. This can speed up large data transfers significantly.
`tcp_conn_req_max_q` `tcp_conn_req_max_q0`	10240	The size of the complete (and incomplete connection) queue. Generally, the default values are sufficient. However, it is recommended to increase these values to 10240. These values can be changed if connection drop problems are observed.
`tcp_slow_start_initial`	4	This setting changes the data transmission rate. Changing this value is important to work arounds to bugs used some operating systems in the implementation of slow start algorithms.

Solaris Kernel Recommendations

To enhance performance, change the Solaris Kernel performance parameters (described in Table 13–3) in the file /etc/system.

Table 13–3 Solaris Kernel Performance Parameters

Parameter	Value	Comment
`rlim_fd_max`	8192	The hard limit for number of file descriptors.
`rlim_fd_cur`	2048	The soft limit for number of file descriptors.

Table 13–3 (Cont.) Solaris Kernel Performance Parameters

Parameter	Value	Comment
`lwp_default_stksize`	0x4000	The LWP stack size.
`rpcmod:svc_run_stksize`	0x4000	The NFS stack size.
`sq_max_size`	1600	By increasing `sq_max_size`, you increase the number of message blocks (mblk) that can be in any given syncq. For every 64mb, add 25 to its value. As a result, the value for 4GB is 1600.

14

Load Balancing and Failover

This chapter, through the following sections, discusses Oracle Application Server Wireless load balancing and failover.

14.1 Overview of Load Balancing and Failover

OracleAS Wireless offers a scalable, reliable server infrastructure through clustering and high availability. The clustering structure includes the following two features.

- Load Balance: mod_oc4j on top of Oracle HTTP Server (OHS) distributes the request workload among multiple OracleAS Wireless server processes.
- Fault Tolerance (Failover): mod_oc4j on top of OHS redirects a client to another working OracleAS Wireless server process if a OracleAS Wireless Server process failure occurs.

14.2 Clustering Architecture

Each OracleAS Wireless Server process which runs on a single Java Virtual Machine (JVM) is referred to as a node. One or more nodes comprise an island. Nodes within an island are capable of serving the same applications, because the session for each client is replicated among all the nodes within an island in preparation of failover. One or more islands together form an OC4J (OracleAS Containers for J2EE) instance for the purpose of load balancing. The entire OC4J instance is linked by mod_oc4j to a simple front-end, Oracle HTTP Server (OHS). Typically, an island has two to four nodes.

By default, the requests from the same client are always redirected to the same OracleAS Wireless Server process. If one process goes down, then the fault tolerance feature is supported for both stateful and stateless requests as follows:

- Stateless Requests – Fault tolerance is achieved by redirecting the client to another working process.
- Stateful Requests – The session state is propagated to the processes within the same island, which enables another process in that same island to pick up the request from a given client if a failover occurs.

14.3 Clustering Configuration

This section describes how to configure the Oracle HTTP Server (OHS), Oracle Process Management and Notification (OPMN), and OracleAS Containers for J2EE (OC4J).

14.3.1 Configuring Oracle HTTP Server (OHS)

The configuration file for OHS is `httpd.conf`, which includes `mod_oc4j.conf`, located in the `ORACLE_HOME/Apache/Apache/conf/` directory. Example 14–1 illustrates how the mounting point from the HTTP request to the OracleAS Wireless Server clustering instance is specified in `mod_oc4j.conf`.

Example 14–1 mod_oc4j.conf

```
LoadModule oc4j_module libexec/mod_oc4j.so
<IfModule mod_oc4j.c>
Oc4jMount /ptg OC4J_Wireless
Oc4jMount /ptg/* OC4J_Wireless
Oc4jMount /mcs OC4J_Wireless
Oc4jMount /mcs/* OC4J_Wireless
Oc4jMount /mcs/media OC4J_Wireless
Oc4jMount /mcs/media/* OC4J_Wireless
</IfModule>
```

When installing the OracleAS Wireless Server from Oracle Universal Installer (OUI), these lines should be automatically populated in the `mod_oc4j.conf` file.

14.3.2 Configuring Oracle Process Management and Notification (OPMN)

The major configuration file for OPMN is *opmn.xml*, located in the ORACLE_HOME/opmn/conf/directory. The definition for the process-type Id `for oc4j_wireless` in `opmn.xml` should be the exactly same as it appears in the mounting specification of the `mod_oc4j.conf (`Example 14–1). The number of islands, the number of processes, and the other configuration parameters are also defined within `opmn.xml`. Example 14–2 illustrates a sample configuration.

Example 14–2 Sample Configurations of Islands and Processes for opnm.xml

```
<ias-component id="wireless" status="enabled">
    <process-type id="OC4J_Wireless" module-id="OC4J">
       <environment>
       ...
       </environment>
       <module-data>
          <category id="start-parameters">
          ...
          </category>
       </module-data>
       ...
       <port id="ajp" range="3301-3400"/>
       <port id="rmi" range="3201-3300"/>
       <port id="jms" range="3701-3800"/>
       <process-set id="OC4J_WirelessIslandA" numprocs="2"/>
       <process-set id="OC4J_WirelessIslandB" numprocs="3"/>
    </process-type>
 </ias-component>
```

For this OC4J_Wireless cluster, two islands (`OC4J_WirelessIslandA` and `OC4J_WirelessIslandB`) share the request workload. `OC4J_WirelessIslandA` is comprised of two wireless server processes while `OC4J_WirelessIslandB` is comprised of three OracleAS Wireless Server processes. Altogether, five ports are needed for each type of protocol. The port number range is from the base-port number to the base-port number plus five. The base-port numbers are dynamically allocated during the installation time.

By default, the OracleAS Wireless Server `<process-set>` element should be populated within `opmn.xml`. However, the populated entry only supports a single OracleAS Wireless Server process and thus is not suitable for load balancing and failover. The configuration for load balancing and failover must be manually added.

14.3.3 Configuring OC4J

The OC4J-related configuration files are located in `ORACLE_HOME/j2ee/OC4J_Wireless/config` directory. The default configuration is set for running single OracleAS Wireless Server process.

To support load balancing and failover features, you must modify the OC4J configuration files `orion-web.xml` and `/WEB-INF/web.xml` as described in the following steps.

1. Modify `orion-web.xml.`

   There are two `orion-web.xml` files, one for Multi-Channel server and one for the wireless mcs applications. They are located in the following directories:

   - `ORACLE_HOME/j2ee/OC4J_Wireless/application-deployments/ptg/ptg-web/`
   - `ORACLE_HOME/j2ee/OC4J_Wireless/application-deployments/mcs/mcs-web/`

   For both of these files, add `<cluster-config />` to the main body of the `<orion-web-app>` tag.

2. Modify /WEB-INF/web.xml

   There are two `web.xml` files, one for OracleAS Wireless web server and one for the mcs modules. They are located in the following directories:

   - `ORACLE_HOME/j2ee/OC4J_Wireless /applications/ptg/ptg-web/WEB-INF/`
   - `ORACLE_HOME/j2ee/ OC4J_Wireless /applications/mcs/mcs-web/WEB-INF/`

   For both of these files, add the `<distributable />` tag to the main body of `<web-app>`

14.4 Configuring OracleAS Wireless for High-Availability Deployment

In Oracle Application Server 10g Release 2 (10.1.2.02), OracleAS Wireless applications cannot be clustered using the Oracle9*i*AS clustering mechanism. However, you can configure Oracle Application Server 10g Release 2 (10.1.2.02) to achieve a high-availability deployment by completing the following steps.

Note: You must back up all files before you modify them.

1. Install the Oracle Application Server 10g Release 2 (10.1.2.02) infrastructure tier on one machine and install multiple middle tiers on separate machines. Ensure that each of these middle-tier installations point to the infrastructure tier.
2. Shut down DCM and all of process by running the command

   `[oracle home]/dcm/bin/dcmctl stop`
3. Shut down Oracle Enterprise Manager (OEM) using the command

   `[oracle home]/bin/emctl stop`
4. Verify that the file `[oracle home]/opmn/conf/ons.conf` exists on each of the mid-tiers. Verify that the infrastructure tier contains IP-address entries for all the other tiers. If not, file and add missing IP-address entries.
5. On each middle tier, increase the number of processes that need to participate in the default island for the OC4J_Wireless OC4J instance to the desired number. This can be done from the Oracle Enterprise Manager Application Server Control or by modifying the file:

   `[oracle home]/opmn/conf/opmn.xml`.

   For details and concepts of OC4J instance and OC4J islands, refer to *Oracle Application Server Containers for J2EE Services Guide*
6. In the `mod_oc4j` configuration file for each middle-tier (that is, `[oracle home]/Apache/Apache/conf/mod_oc4j.conf`), modify the mount-point entries for the OracleAS Wireless runtime. These entries should be of the form illustrated in Example 14–3 and must be the same for all middle- tier machines.

Example 14–3 Mount-Point Entries in mod_oc4j.conf

```
Oc4jMount /ptg instance://c.host4CPU.mysite.com:OC4J_Wireless,
c.host2CPU.mysite.com:OC4J_Wireless

Oc4jMount /ptg/* instance://c.host4CPU.mysite.com:OC4J_Wireless,
c.host2CPU.mysite.com:OC4J_Wireless

Oc4jMount /mcs instance://c.host4CPU.mysite.com:OC4J_Wireless,
c.host2CPU.mysite.com:OC4J_Wireless

Oc4jMount /mcs/* instance://c.host4CPU.mysite.com:OC4J_
Wireless,c.host2CPU.mysite.com:OC4J_Wireless

Oc4jMount /mcs/media instance://c.host4CPU.mysite.com:OC4J_
Wireless,c.host2CPU.mysite.com:OC4J_Wireless

Oc4jMount /mcs/media/* instance://c.host4CPU.mysite.com:OC4J_
Wireless,c.host2CPU.mysite.com:OC4J_Wireless

Oc4jSelectMethod roundrobin:weighted
Oc4jRoutingWeight host4CPU.mysite.com 2
Oc4jRoutingWeight host2CPU.mysite.com 1
```

Note: In Example 14–3, c represents the instance name.

7. Run `[oracle home]/dcm/bin/dcmctl updateConfig` to update the DCM repository with the configuration file changes.

On slow machines, a DCM error (timeout) of the form ADMN-906005 may appear. If this occurs, run the command `[oracle home]/dcm/bin/dcmctl getReturnStatus` and wait until the command exits. This confirms that the changes have been propagated to the DCM repository.

8. Add the `<cluster-config/>` tag under the `<orion-web-app>` tag in `[oracle home]/j2ee/OC4J_wireless/application-deployments/ptg/ptg-web/orion-web.xml`.
9. Start DCM and all processes by running the command

   `[oracle home]/dcm/bin/dcmctl start`.
10. Start Enterprise Manager by running the command

    ```
    [oracle home]/bin/emctl start
    ```
11. Configure a hardware load-balancer to point to the middle-tiers.

Currently, high-availability support is only available for the core server runtime (by default mapped to the URI `/ptg/rm`).

For more information, refer to the *Oracle Application Server Containers for J2EE User's Guide* .

15 Globalization

This chapter includes the following sections:

15.1 Overview of Globalization

Oracle Application Server Wireless supports multi-locale and multi-encoding. The OracleAS Wireless Server dynamically determines locale and request and response encoding based on the runtime context.

15.2 Determining a User's Locale

The OracleAS Wireless Server dynamically determines the appropriate locale of a user by using such locale information as PALocale, the user's preferred locale, the Accept Language header, and the site locale.

`PAlocale` is a HTTP parameter that specifies the preferred value before login. The possible value for the `PAlocale` parameter follows the HTTP accept-language header format. For example, `PAlocale` = en-US. This format is distinct from the Java locale format (`en_US`).

The user's preferred locale is the language preference of a OracleAS Wireless user, which is set with the User Manager. For more information, see Section 5.5.

The Accept Language header is an HTTP protocol parameter that user agents (Web browsers) send with HTTP requests.

> **Note:** For information on the HTTP accept-language header format, see the HTTP specification of the World Wide Web Consortium (W3C).

The Site Locale is an instance-wide default locale of the OracleAS Wireless Server. For more information, see Section 15.2.4.

15.2.1 After Login

After login, the OracleAS Wireless Server respects the user's preferred locale.

15.2.2 Before Login

Before login, the OracleAS Wireless and Voice Portal (ptg/rm), Async Listener, the OracleAS Wireless Tools and the Customization Portal each determine the appropriate locale of a user's device.

Table 15–1 illustrates how the Async Listener, the OracleAS WirelessWeb Server, the OracleAS Wireless Tools and the Customization Portal determine the locale of a user. The numeric value indicates the preference for the detection methods in descending order.

Table 15–1 Locale Determination

Method	Async Listener	OracleAS Wireless and Voice Portal	OracleAS Wireless Tools and Customization Portal
Locale of the registered user or virtual user	1	1	1
HTTP parameter: PAlocale	N/A	2	N/A
Accept-language http header	N/A	3	N/A
Site default locale	2	4	2

15.2.2.1 OracleAS Wireless Wireless and Voice Portal

The OracleAS Wireless and Voice Portal (ptg/rm) determines the locale of a user in the following order:

1. Use `PAlocale` (if present).
2. Use the Accept_Language HTTP header (if present).
3. Use the site default locale.

15.2.2.2 The OracleAS Wireless Tools and Customization Portal

The OracleAS Wireless Tools and Customization Portal determine the location of a user in the following order:

1. Use `PAlocale` (if present).
2. Use the site default locale.

15.2.2.3 Async Listener

The Async Listner determines the location of a user in the following order:

1. Use the user's preferred locale if the connecting user can be identified through the device ID.
2. Use the site default locale. For more information, see Section 15.2.4.

15.2.3 Setting the Locale for a User Profile

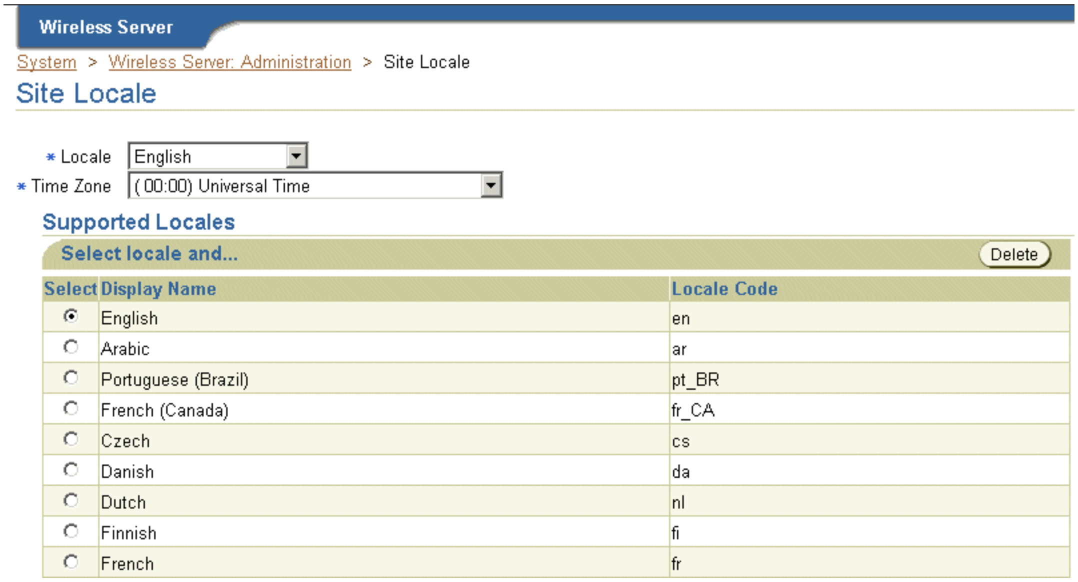
You can set a preferred location for a user when you create a user or edit a user profile. If the preferred location is not specified, then the default site locale is used.

15.2.4 Setting the Site Locale

From the *Site Administration* page of the System Manager (accessed through the Oracle Enterprise Manager Application Server Control), you can specify the default site locale and add to the list of locales that the site can support. Use a java locale (such as en_US) when adding to the list of supported locales (depicted in Figure 15–1). For more information, see Section 3.9.1.4.

> **Note:** You can also set the site locale using the *Basic Site Configuration* wizard, accessed from the *Home* page of the System Manager. For more information, see Section 3.3.

Figure 15–1 The Site Locale Screen of the System Manager (Partial View)



System > Wireless Server: Administration > Site Locale

Site Locale

* Locale: English

* Time Zone: (00:00) Universal Time

Supported Locales

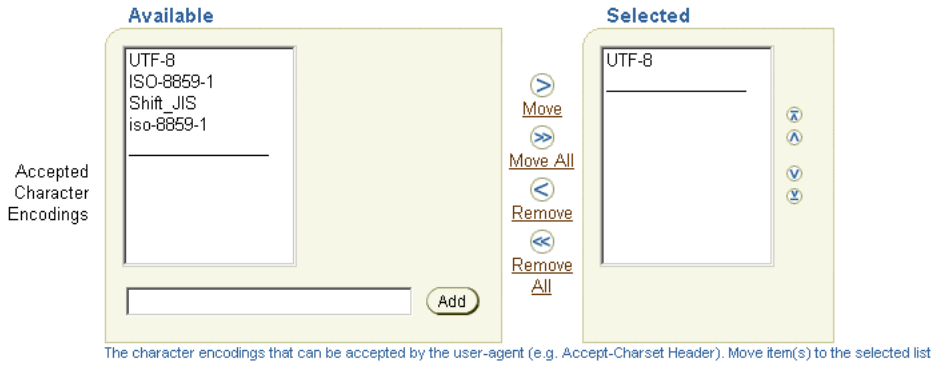
Select locale and...

Select	Display Name	Locale Code
●	English	en
○	Arabic	ar
○	Portuguese (Brazil)	pt_BR
○	French (Canada)	fr_CA
○	Czech	cs
○	Danish	da
○	Dutch	nl
○	Finnish	fi
○	French	fr

15.3 Determining the Encoding of a Device

The content sent to the device is encoded using the character encoding of the device. The character encoding is among the device attributes stored in the OracleAS Wireless repository. Using the Foundation Manager, you can edit the browser capabilities of a device in the OracleAS Wireless repository to update it to the encoding appropriate to a given country or locale (Figure 15–2). For more information on creating, cloning, or editing a device, see Section 8.3.2, Section 8.3.3, and Section 8.3.2.1, respectively.

Figure 15–2 Editing the Encoding for a Device (Partial View of the Editing Function)



Available: UTF-8, ISO-8859-1, Shift_JIS, iso-8859-1

Selected: UTF-8

The character encodings that can be accepted by the user-agent (e.g. Accept-Charset Header). Move item(s) to the selected list

The Internet Assigned Numbers Authority (IANA) specifies the valid values for the character encoding format of a device. The names of the IANA character set are available at:

http://www.iana.org/assignments/character-sets

Table 15–2 illustrates how the encoding is determined.

Table 15–2 Determining the Device Encoding at Runtime

Component	Method of Determination
Multi-Channel Server	1. Using the Accept-Charset HTTP header sent by the device. 2. If the device does not send the Accept-Charset HTTP header, then the value stored in the OracleAS Wireless repository is used instead.
Async Listener	Determined by the corresponding transport driver.
OracleAS Wireless Tools and Customization Portal	1. Using the Accept-Charset HTTP header sent by the device. 2. If the device does not send the Accept-Charset HTTP header, then the character encoding of the device called PAPZ is used instead. The default encoding is UTF-8.
Notification Application	Determined by the corresponding transport driver.

15.3.1 HTTPAdapter – Based Service

This section describes the encoding for the request and response of a HTTPAdapter-based application

15.3.1.1 Encoding for the Request of an HTTPAdapter-Based Application

When sending the HTTP request to the remote content provider, only the parameters of the HTTPAdapter application are encoded using the `input_encoding` of the application (if it is specified). Use the encoding format of the IANA (Internet Assigned Numbers Authority) when specifying the value for `input_encoding`.

15.3.1.2 Best Practice for Writing Multi-Channel Applications Using JSPs.

When using JSPs to write Multi-Channel applications, insert the content described in Example 15–1 at the top of the JSP.

Example 15–1 Multi-Channel Content

```
XHTML MP Content:
```

```
<%
  String userAgent = request.getHeader("User-Agent");
  if(userAgent != null && userAgent.indexOf("PTG/2.0 (Oracle9iAS Wireless") >= 0)
{
    response.setContentType ("application/vnd.wap.xhtml+xml; charset=UTF-8");
%><?xml version = "1.0" encoding = "UTF-8" standalone="yes" ?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
  "http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
<%
} else {
  response.setContentType ("text/xml; charset=UTF-8");
%><?xml version = "1.0" encoding = "UTF-8" standalone="yes" ?>
<%
}
  // Set the request character encoding
  String encoding = request.getCharacterEncoding();
  if(encoding == null) {
    encoding = request.getHeader("x-oracle-mcs.character.encoding");
    encoding = ((encoding == null) || (encoding.length() == 0)) ? "UTF-8" :
encoding;
    request.setCharacterEncoding(encoding);
  }

  // Optional - Prevent Page Caching
  response.setHeader("Cache-Control", "no-store"); // HTTP 1.1
  response.setHeader("Pragma", "no-cache"); // HTTP 1.0
  response.setHeader("Expires", "0"); // prevents caching at the proxy server
%>


MobileXML Content:

<%
  String userAgent = request.getHeader("User-Agent");
  if(userAgent != null && userAgent.indexOf("PTG/2.0 (Oracle9iAS Wireless") >= 0)
{
    response.setContentType ("text/vnd.oracle.mobilexml; charset=UTF-8");
%><?xml version = "1.0" encoding = "UTF-8" standalone="yes" ?>
<!DOCTYPE SimpleResult PUBLIC "-//ORACLE//DTD SimpleResult 1.1//EN"
"http://xmlns.oracle.com/ias/dtds/SimpleResult_1_1_0.dtd">
<%
} else {
  response.setContentType ("text/xml; charset=UTF-8");
%><?xml version = "1.0" encoding = "UTF-8" standalone="yes" ?>
<%
}
  // Set the request character encoding
  String encoding = request.getCharacterEncoding();
  if(encoding == null) {
    encoding = request.getHeader("x-oracle-mcs.character.encoding");
    encoding = ((encoding == null) || (encoding.length() == 0)) ? "UTF-8" :
encoding;
    request.setCharacterEncoding(encoding);
  }

  // Optional - Prevent Page Caching
  response.setHeader("Cache-Control", "no-store"); // HTTP 1.1
  response.setHeader("Pragma", "no-cache"); // HTTP 1.0
  response.setHeader("Expires", "0"); // prevents caching at the proxy server
```

```
%>
```

Note: This example uses UTF-8 encoding. Substitute this value for that of the correct encoding for the application. Be sure to replace all occurences of this value.

15.3.1.3 Encoding for the Response of an HTTP Adapter-Based Application

OracleAS Wireless determines the encoding of the response of an HTTPAdapter-based application in the following order:

1. Charset as part of the content-type header on the response.
2. Input-encoding (if present) of the input parameter of the application.
3. ISO-8859-1 (the default).

15.4 Languages Available for Online Help

Users can view the online help for the OracleAS Wireless Tool and the Customization Portal in 29 languages. The site locale, configured through the System Manager, determines the display language. For more information, see Section 3.9.1.4.

In this release, the built-in labels and on-line help for the OracleAS Wireless Tools and System Manager display in nine languages.

OracleAS Wireless and Voice Portal (ptg/rm) can display the built-in labels in 29 different languages.

15.5 Driver Encoding

Each driver handles encoding individually .

16

Integrating OracleAS Wireless with Other Components

The chapter includes the following sections:

- Section 16.1, "Overview of Integrating OracleAS Wireless with OID and Portal"
- Section 16.2, "Integrating OracleAS Wireless with Oracle Application Server Portal"
- Section 16.3, "Notification Engine Integration"
- Section 16.4, "Implementing Virtual Private Portals on OracleAS Wireless"

16.1 Overview of Integrating OracleAS Wireless with OID and Portal

This chapter describes integrating OracleAS Wireless with the Oracle Application Server components, Oracle Internet Directory (OID) and OracleAS Portal. In this release, user information is stored centrally in OID. The SSO (Single Sign-On) server uses an OID repository to authenticate users. Table 16–1 describes the attribute mapping between PanamaUser (stored in Oracle Application Server Wireless repository) and orclUserV2 user attributes (stored in OID).

Table 16–1 Attribute Mapping between PanamaUser and orclUserV2 user

PanamaUser	OID User
Name	orclcommonnicknameattribute (by default, *cn*) specified in OID configuration
DisplayName	DisplayName
Enabled	orclIsEnabled
PasswordHint	orclPasswordHint
PasswordHintAnswer	orclPasswordHintAnswer
Language and Country	preferredLanguage
TimeZone	TimeZone
DateofBirth	orclDateOfBirth
Globaluid	orclguid (the orclguid attribute uniquely identifies OID Users)
Password	user password
Password Confirm	Confirms user password.
Gender	orcl header

Administrators use tools such as Delegated Administrative Services (DAS), to create a new user in OID or to modify attributes of an existing user. Alternatively, OracleAS Wireless customers can implement their own user administrator tool to create, modify, or delete users with the OracleAS Wireless model APIs.

The user information is synchronized between OracleAS Wireless and OID repositories using the following mechanisms:

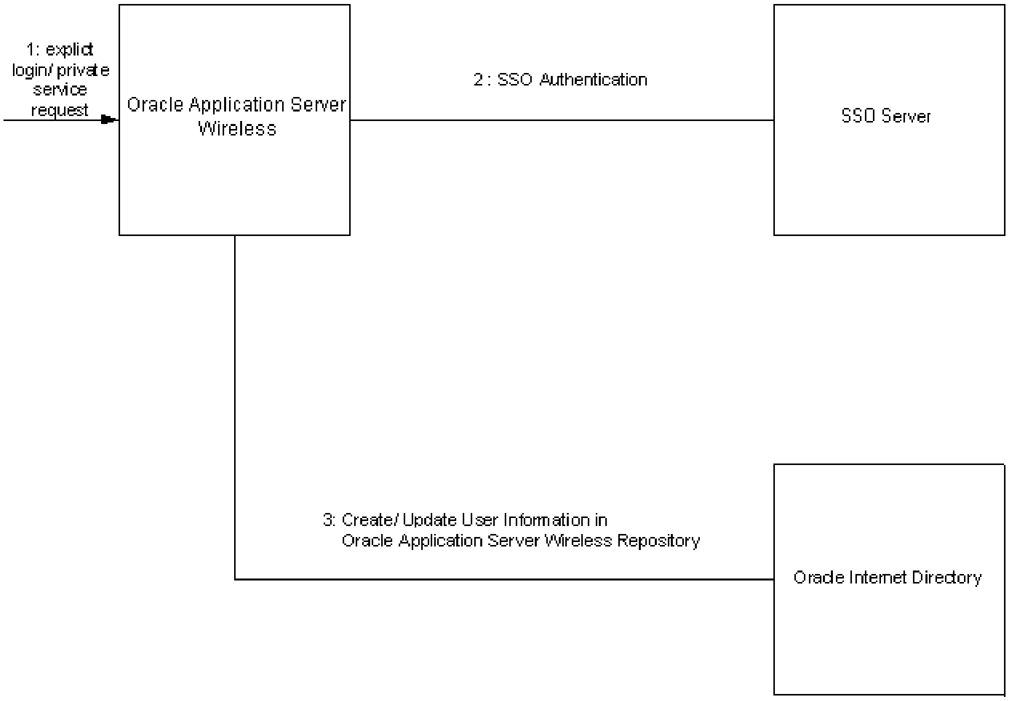
- Oracle Application Server Wireless repository synchronization after user authentication
- PL/SQL based asynchronous synchronization
- Oracle Application Server Wireless model API interface

For information on authenticating users through SSO, see Chapter 11, "Mobile Single Sign-On".

16.1.1 Repository Synchronization after User Authentication

OracleAS Wireless synchronizes user information which is stored in the Wireless repository with OID after SSO authentication.

Figure 16–1 Interactions Between Oracle Application Server Wireless, SSO and OID


1: explicit login/ private service request
Oracle Application Server Wireless
2 : SSO Authentication
SSO Server
3: Create/ Update User Information in Oracle Application Server Wireless Repository
Oracle Internet Directory

The authentication sequence (as depicted in Figure 16–1) is as follows:

1. A user sends an explicit login request or tries to access a private service, or an external SSO partner application.
2. The SSO server challenges user credentials and the user is authenticated.
3. If the authenticated user does not exist in the OracleAS Wireless repository, then OracleAS Wireless retrieves the user information from OID and creates a new user in the OracleAS Wireless repository. Otherwise, the User attributes in the local repository are synchronized with the attributes stored in the OID.

Note: The user attributes must be synchronized with OID because the PL/SQL notification mechanism does not guarantee real-time notifications.

16.1.1.1 Potential Conflicts in Application Entities Based on OID

The OracleAS Wireless middle tiers installed against the common meta data repository (the Oracle Application Server Wireless schema) share a common application entity in OID. The application entity is created as part of the first OracleAS Wireless middle-tier installation and is owned by the OID user who installs that middle tier. Subsequent OracleAS Wireless middle tiers installed against the same meta data repository use the application entity that was create as part of the first middle tier installation.

Subsequent OracleAS Wireless middle-tier installations against a meta data repository should be done by the same OID user who installed the first OracleAS Wireless middle tier. For a different OID user to add other OracleAS Wireless middle tiers, you must add the OID user as a shared owner of the application entity before stating any subsequent OracleAS Wireless middle-tier installations.

To add a shared owner for an OracleAS Wireless application entity:

1. Find the name of the OracleAS Wireless application by executing

   ```
   ORACLE_HOME/wireless/bin/getAppEntityName.sh[bat]
   ```

   from the first middle tier. This script prints the name of the OracleAS Wireless application entity.

2. Use the OID Deployment Console or OID Directory Manager to add the new OID user as a Component Owner for the OracleAS Wireless application entity name returned in the previous step.

16.1.2 PL/SQL-Based Asynchronous Synchronization

The Oracle Application Server Wireless installation registers a PL/SQL procedure with OID. The PL/SQL procedure is invoked when a user is modified or deleted in OID.

Figure 16–2 Interactions between PL/SQL and OID

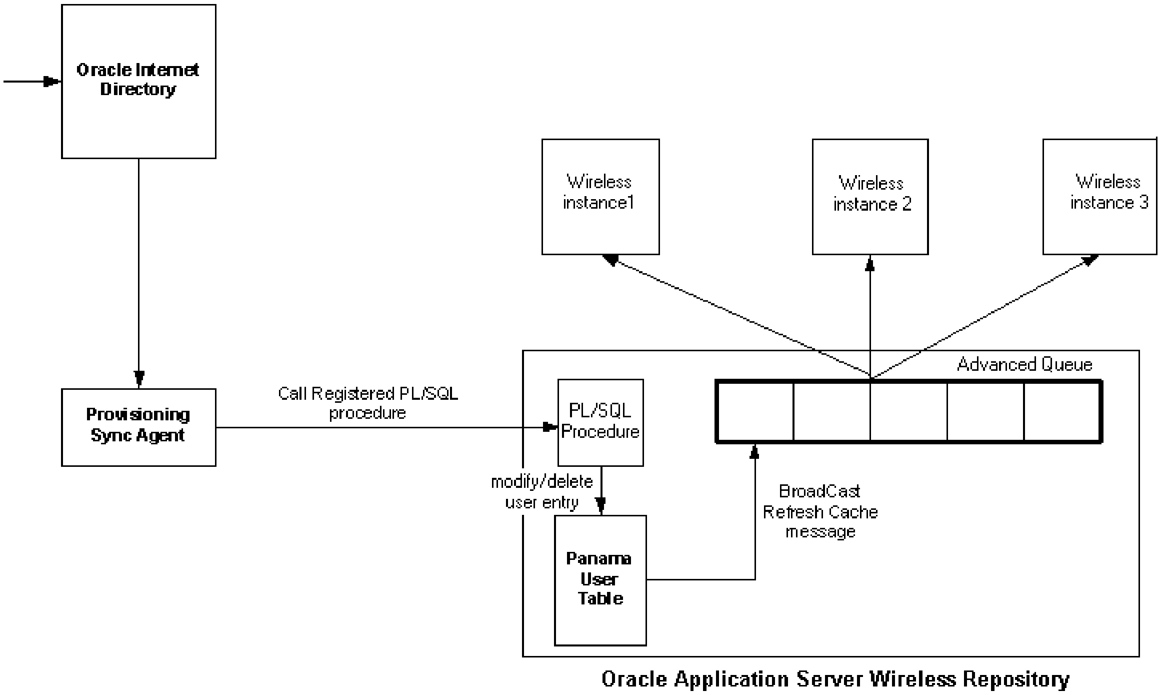


Figure 16–2 depicts the events triggered when a user is modified in OID. The sequence is as follows:

1. A user attribute is modified, or the user is deleted in OID.
2. The Provisioning Synchronization agent picks up the modifications and calls the registered PL/SQL package.
3. The PL/SQL package accomplishes appropriate changes in the PanamaUser table (if required).
4. The trigger on the PanamaUser table broadcasts a RefreshCache message to all running instances of OracleAS Wireless.
5. If the modified PanamaUser is cached by the running instances, the PanamaUser object is reloaded from the OracleAS Wireless repository.

16.1.3 Oracle Application Server Wireless Programmatic Model API Interface

The `ModelFactory.createUser()` method creates a corresponding user in the OID repository.

The `User.set` methods update the corresponding user entry in OID for all of the attributes. The `User.delete()` method removes the corresponding user from the OID repository. The current semantics of commit is preserved for the user modifications.

16.1.4 OracleAS Wireless User Management Integrated with DAS

In OracleAS Wireless integration mode, when you create a user through the User Manager, the request is first redirected to OID DAS (Delegated Administration Service), for entering Oracle Application Server User Common Attribute Values. After that, the request is redirected back to the User Manager page for entering Wireless-specific attribute values.

The same applies for editing a registered Wireless user. The user is first edited through DAS and then through the User Manager.

16.1.5 Synchronizing Data between Oracle Application Server and Oracle Internet Directory

To synchronize data between Oracle Application Server and OID, run the Oracle Directory Integration Server, `odiserv`.

16.2 Integrating OracleAS Wireless with Oracle Application Server Portal

Oracle Application Server Portal (Portal) is a Web-based application model for building and deploying e-business portals. It provides an environment for accessing and interacting with enterprise software services and information resources. Portal provides a framework that integrates Web-based resources such as Web pages, applications, business intelligence reports, and syndicated content feeds, within standardized, reusable information components called portlets. For more information, see *Oracle Application Server Portal Configuration Guide*.

A portlet is an area of HTML/XML located within a defined area of a Web page. Portlets communicate with the portal through an entity called a provider. Portlets form the fundamental building blocks of a Portal page. Each Portal page consists of content presented through one or more portlets and links that enable the user to navigate to another page to take some action.

Portlets summarize, promote, or provide basic access to an information resource. The portlets allow information resources to be personalized and managed as an application of Portal. The portal framework provides additional services including SSO (single sign-on), content classification, enterprise search, directory integration, and access control. Portal supports Desktop PC Web browsers and enables access to Portal pages from wireless devices. Portal, working in conjunction with OracleAS Wireless, automatically transforms the Portal page structure to one that befits the wireless devices. Portlets provide a wireless interface using Portal through OracleAS Wireless, because Portal generates the page structure in OracleAS Wireless XML for all requests from wireless devices that OracleAS Wireless renders to the device.

16.2.1 OracleAS Portal as a Wireless Application

Portal must be deployed as a OracleAS Wireless application in the OracleAS Wireless repository to enable OracleAS Wireless access to Portal. Each Portal installation is deployed as an HTTP Adapter-based application in OracleAS Wireless. Multiple Portals may be deployed on a single OracleAS Wireless instance. The HTTP adapter application, which accepts a URL as a configuration parameter, must be set to the URL of the Portal's home page. To create a OracleAS Wireless application, a master application definition based on an HTTP adapter must be created using the Service Manager. In addition, you must create aPortal application based on the HTTP adapter master application.

Portal redirects requests from a mobile device to an OracleAS Wireless server. The OracleAS Wireless Server accepts the request and invokes the Portal home page over HTTP and accepts the response generated (in OracleAS Wireless XML), from Portal. The Wireless XML response, generated by Portal, is then adapted to the native device markup by the OracleAS Wireless server. All further requests and responses between Wireless device and OracleAS Portal is mediated by the OracleAS Wireless Server.

The mobile devices make the first request to Portal server. Portal redirects the device request to OracleAS Wireless Server. The Portal appends two query parameters to the redirected URL, `PAoid` and `PAhome`. Both `PAoid` and `PAhome` contain the value of the object id (the service-id in the OracleAS Wireless repository) of the Portal's HTTP adapter service. The syntax of the redirected URL is:

```
http://9iASWSerrver:port/ptg/rm?PAoid=<OraclePortal object
id>&PAhome=<OraclePortal object id>
```

The *PAoid* parameter enables the Wireless server to directly launch the Portal Home page, without having to navigate through the Wireless server's folder and service hierarchy. The `PAhome` parameter sets the Portal's *Home* page as the home page for the current OracleAS Wireless session.

16.2.2 Developing Wireless Portlets

Portlets are owned by entities called providers. One provider can manage one or many portlets. Providers are the backbone of the portlets displayed on each page. Portal supports a Web Provider framework that is written as a Web application. It is installed and hosted on a Web server and is remote from the Portal. A portlet exposed as a Web Provider can be developed in any Web language. A Web Provider communicates with Oracle Application Server Portal using SOAP(XML).

Portal supports a Java-based Portal Developer Kit (PDK) framework to develop portlets and services. The Java PDK framework is a set of services that enable Java programmers to create portlets from existing Java-based applications (Java, Java Servlets, and JSPs). It provides an abstraction to handle communication with Oracle Application Server Portal, default classes to simplify portlet creation, and exposes APIs for end-user customization, session storage, security, and logging.

For mobile devices, Portal supports portlets that generate OracleAS Wireless XML. To enable wireless access, Portlets must generate OracleAS Wireless XML and indicate this capability using the Java PDK framework. The Java PDK framework uses a Provider.xml file to discover the capabilities of the Portlets supported by a Provider. Refer to Oracle Portal's PDK-Java User's Guide for more information.

16.2.2.1 Provider.xml Tags

This section provides an overview of the tags in the Provider.xml file that indicate the wireless capabilities of a portlet.

<acceptContentType>

Usage:
`<acceptContentType>text/vnd.oracle.mobilexml</acceptContentType>`

The text/vnd.oracle.mobilexml value indicates that the portlet generates the OracleAS Wireless XML required for wireless access. A portlet can be enabled for both HTML (PC Desktop) and wireless access by indicating that it can accept both of these content types as:

```
<acceptContentType>text/vnd.oracle.mobilexml</acceptContentType>
            <acceptContentType>text/html</acceptContentType>
```

If the portlet generates only OracleAS Wireless XML (text/vnd.oracle.mobilexml), then the portlet transforms the OracleAS Wireless XML to HTML for PC Desktop clients unless otherwise indicated.

<mobileFlags>

Usage: `<mobileFlags>MOBILE_ONLY</mobileFlags>`

Portlets set this value to `MOBILE_ONLY` to indicate that this portlet must be renderedon wireless devices only. This prevents the default behavior of a portal that renders portlet-generated OracleAS Wireless XML to PC Desktop clients.

<showLink>

Usage: `<showLink>true</showLink>`

Portal renders all of the portlets on mobile devices as links. Portlets must set this value to true to render portlets on a wireless device. The true value enables the portal to generate a link, which points to the portlet content on the wireless device.

<linkPage>

Usage:

```
<linkPage class="oracle.portal.provider.v2.render.http.ResourceRenderer">
              <resourcePath>/mypath/mypage.jsp</resourcePath>
              <contentType>text/vnd.oracle.mobilexml</contentType>
              </linkPage>
```

The <linkPage> tag holds the pointer to the resource which generates the required link that is rendered on a wireless device. This resource must generate OracleAS Wireless XML. Example 16–1 illustrates a link page implemented in JSP.

Example 16–1   A Link Page Implemented in JSP

```
<%@ page session="false" contentType="text/vnd.oracle.mobilexml" %>
    <SimpleHref target="/mypath/mywireless.jsp" label="Go">
          Wireless HelloWorld
    </SimpleHref>
```

The new version JPDK has been updated to understand thee wireless properties of a Portlet. The JPDK also supports Wireless-specific request information such as location and device information, which can be accessed by the Portlets through the JPDK APIs.

16.2.3 Oracle Portal, OracleAS Wireless and Single Sign-On (SSO)

Both Oracle Portal and OracleAS Wireless depend on Oracle's SSO solution for user authentication and login. This integration enables the user to invoke protected applications defined on both systems and eliminates multiple login dialog boxes for users.

OracleAS Wireless Server upgrades the session context of a user to an "authenticated" state when any service or application (HTTPAdapter applications) validates the user credentials with the SSO server. When Portal, a mobile application, validates the credentials of a user with the SSO Server, the session context in OracleAS Wireless is also updated.

16.2.4 Portlets for Applications Deployed on Wireless Server

Portal’s applications provide a PC Desktop view of OracleAS Wireless services. Use the Portal JPDK framework to provide a “showPage" and "editPage", for Web-based customizations.

Since the Portal itself can be accessed from a wireless device, you must also provide a mobile portlet. On a wireless device, the mobile portlets are rendered as links and can be made to point to an application deployed on the OracleAS Wireless server. Use Portal's JPDK framework to provide a “linkPage" that generates the appropriate link to the OracleAS Wireless service. To point to a OracleAS Wireless service from a mobile portlet, use following URL syntax in the OracleAS Wireless XML:

```
target="___REQUEST_NAME__?___SESSION__&amp;PAoid=<PAoid of Wireless Service>"
```

The OracleAS Wireless server replaces all "_<Name>_" to the correct values at runtime and invokes the application defined in the OracleAS Wireless repository.

Example 16–2 illustrates a sample link page.

Example 16–2 A Sample Link Page

```
<%@ page session="false" contentType="text/vnd.oracle.mobilexml" %>
          <SimpleHref target="/___REQUEST_NAME__?PAoid="+PAoid + "&amp;___SESSION_
_" label="Go">
                My Wireless Service
          </SimpleHref>
```

Mobile devices make the first request to Portal server. Portal redirects the device request to OracleAS Wireless server, over HTTP, and appends query parameters to the redirected URL, *PAoid* and *PAhome*. Both *PAoid* and *PAhome* contain the Portal's object and service ID. The typical syntax of the redirected URL is:

`http://`Oracle Application Server Wireless`Serrver:port/ptg/rm?PAoid=<OraclePortalServiceid>&PAhome=<OraclePortalService id>`

The PAoid parameter enables the OracleAS Wireless Server to directly launch the Portal Home page without having to navigate through the OracleAS Wireless Server's folder and service hierarchy. The *PAhome* parameter sets the Portal's Home page as the home page for the current OracleAS Wireless session.

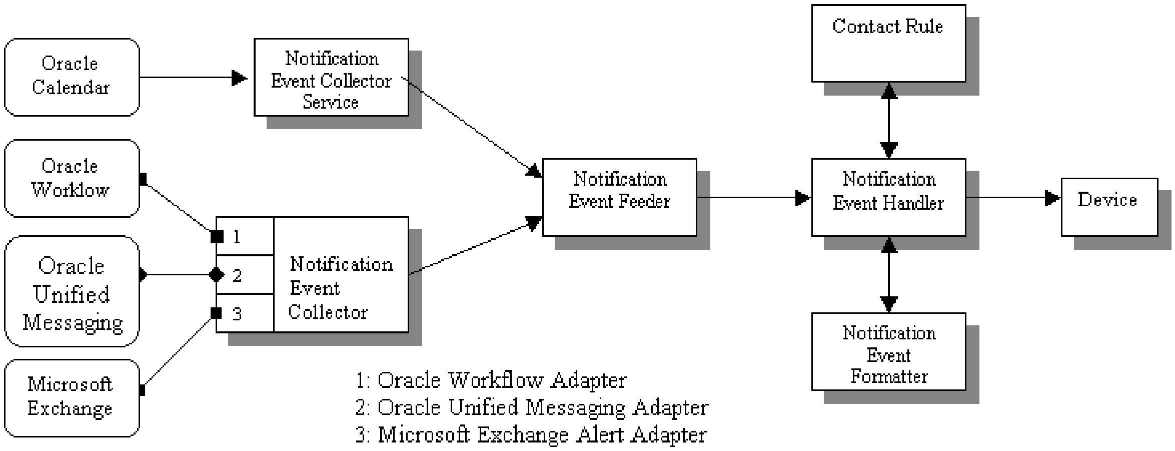
16.2.4.1 OracleAS Wireless Tools and Customization as Portal Providers

The post-installer automatically registers the OracleAS Wireless Tools and Customization as two Portal providers. Thus, if a Portal user selects the two providers, the user then sees two portlets: one for the OracleAS Wireless Tools, and one for Customization. If the URL for Tools or Customization is changed, then the provider can be registered from Wireless System Manager (a node of Oracle Enterprise Manager).

16.3 Notification Engine Integration

The Notification Event Collector Service process uses the OracleAS Wireless Notification Engine to deliver notifications to mobile devices. It adds components that collect application events, process user contact rules, and formats notification contents. Figure 16–3 presents an architectural overview of the various components of the notification process.

Figure 16–3 Integrated Notification Solutions


Oracle Calendar
Oracle Workflow
Oracle Unified Messaging
Microsoft Exchange
Notification Event Collector Service
1
2
3
Notification Event Collector
Notification Event Feeder
Contact Rule
Notification Event Handler
Notification Event Formatter
Device
1: Oracle Workflow Adapter
2: Oracle Unified Messaging Adapter
3: Microsoft Exchange Alert Adapter

Applications outside of OracleAS Wireless can use two different mechanisms to interface with the Notification Engine: the push interface and the pull interface.

Using the push interface, applications send notification events over HTTP to the Notification Event Collector Service which is based on a servlet. The Notification Event Collector Service then passes the notification event data to the Notification Event Feeder, which is a customized data feeder to the Notification Engine.

The pull interface enables the Notification Event Collector process to connect to the application and retrieve the notification events. The notification event data is then passed onto the Notification Event Feeder. The Notification Event Collector process consists of a number of different adapters; each adapter is specific for a particular application. You can enable and disable adapters by configuring the notification collector process. Using the OracleAS Wireless System Manager, you can start or stop a Notification Event Collector process. For more information, see Section 3.6.1.

The notification event handler is a customized system-level notification application that reads data from the notification event feeder. The data indicates the target user for this notification, as well as the type of notification and other notification-specific data.

The notification event handler then looks up the target user’s active contact rule to determine the user’s preferred notification device type and address. The notification event formatter is then invoked, which generates the content of the notification, customized for the user’s device type. The generated notification content is delivered to user’s devices by the notification engine.

The notification event handler is a system-level notification application; users do not need to explicitly create a notification subscription on this process to receive notifications. Instead, only the administrator user, *ORCLADMIN*, is subscribed to this process. Depending on the application, users can specify (either in the OracleAS Wireless Customization Portal, or in the actual application itself), the events for which they want to receive notifications. For each notification processed, the system looks up the contact rule of the target user and make sure that the correct user receives the notification. Use the System Manager to start, stop or configure notification event process. For more information, see Section 3.6.1.

Note: Only the *ORCLADMIN* user can subscribe to the notification event handler notification application. If there is more than one subscription, then users will receive multiple copies of each notification (as many copies as there are subscriptions to the notification event handler notification application).

The Notification Event Collector and notification event handler are two separate processes. Both of them must be running at the same time for the system to process application event notifications.

16.3.1 Integrating OracleAS Wireless with Oracle Workflow

Oracle Workflow integration includes two components: a *notification service* which receives notifications from the Oracle Workflow Notification queues and sends them to the user's mobile device and an *Oracle Workflow Notification Worklist* service which can be accessed through the OracleAS Wireless portal.

Since Oracle Workflow and OracleAS Wireless are both components of Oracle Application Server, OracleAS Wireless connects to Oracle Workflow through OID. As a result, they share the same user repository.

16.3.1.1 Notification Service

Oracle Workflow provides a queue which contains all of the outgoing notifications for that particular instance. Each message in the queue contains all of the necessary information for the notification and for the user to which it is sent. OracleAS Wireless dequeues these messages and uses XMS to construct a message to be sent to the end user. The user can then respond to this notification. The response is directed to a OracleAS Wireless service which will then update Oracle Workflow according to the user's response.

Note: If end users cannot receive notifications during the testing of the OracleAS Wireless integration with Oracle Workflow, then you must check the log file for an ORA-4031 error, which indicates that the notification service failed because of insufficient memory pool size in the database. To increase the shared memory pool:

1. Increase the value for the *shared_pool_size* parameter in the init.ora file. (Typically, the init.ora file is located on the infrastructure machine in the ORACLE_HOME/dbs directory.)
2. Restart the database for the change to take effect.

If end users still cannot receive notifications, then you must further increase the size of the shared memory pool.

16.3.1.2 Worklist Service

This is the equivalent of the Oracle Workflow Notification Worklist through the OracleAS Wireless portal. Using OID, the Worklist Service will connect to Workflow to retrieve a list of all the user's open notifications. Each notification can be closed or responded to (depending on the type of notification).

16.4 Implementing Virtual Private Portals on OracleAS Wireless

To implement virtual private portals (VPP), uncomment the `ENABLE MULTIPLE REALM SUPPORT` section of the OracleAS Wireless `loginpage.jsp` (Example 16–3).

Example 16–3 The OracleAS Wireless loginpage.jsp

```
<!--
UNCOMMENT TO ENABLE MULTIPLE REALM SUPPORT
NOTE: Please replace "Company 1", "Company 2" with the real values.
  Add as many SimpleFormOption elements as you need
      <SimpleFormSelect name="subscribername" displaymode="list" multiple="false">
        <SimpleTitle>
        <%=utils.getString("visual.login.subscribername", locale)%>
        <SimpleFormOption value="Company 1"selected="true"Company
1</SimpleFormOption>
        <SimpleFormOption value="Company 2">Company 2</SimpleFormOption>
      </SimpleFormSelect>
-->
```

For more infomation on VPP, refer to the *Oracle Application Server Portal Configuration Guide*

A

Troubleshooting Oracle Application Server Wireless

This appendix describes common problems that you might encounter when using Oracle Application Server Wireless and explains how to solve them. It contains the following topics:

- Section 16.5, "Problems and Solutions"
- Section 16.6, "Diagnosing OracleAS Wireless Server Problems"
- Section 16.7, "Viewing UTF-8 Pages in Localized Languages with Netscape 4.7 or Earlier"
- Section 16.8, "Oracle Workflow and Oracle Application Server Wireless"
- Section 16.9, "Re-Registering the OracleAS Wireless Portal Services URL Reference in OracleAS Portal"
- Section 16.10, "Information on HTTPS Support"
- Section 16.11, "Configuring OracleAS Wireless for Load-Balancing"
- Section 16.12, "Need More Help?"

16.5 Problems and Solutions

This section describes common problems and solutions. It contains the following topics:

- Deploying XHTML+XForms, XHTML MP, and SimpleResult Documents
- TON and NPI Parameters
- Configuring SMPP Parameters
- Using Third-Party Authenticators with OracleAS Wireless
- Problem Resetting Passwords
- Postinstallation Configuration
- Error Enqueuing Received Messages
- Errors in Receiving Messages
- Setting the Proxy in XMS
- Unable to Reach XMS Gateway
- Dialing Country Codes with SMS or MMS

- Bad Username/Password Exceptions
- Error When Checking Message Status
- AddressData Class and Message Sending Failure
- Unable to Get Delivery Status Callback
- Slow Performance After Deploying Applications
- Finding Oracle Sensor Edge Server Information and Downloads
- Intermittent Browser Crashes
- Incorrect Display of Non-ASCII Characters on Some Devices
- Service Error Caused by HTTPAdapter Error
- Downloading SSL Root Certificates
- Setting Up OracleAS Wireless to Support HTTPS
- Out of Memory Exceptions in Log Files
- Non-ASCII User Name Corrupted in HDML Browser
- Configuration Assistant Hanging
- DTMF Attribute <SimpleMenu>
- Errors When Starting OracleAS Wireless Using the Oracle Process Management and Notification Utility (opmnctl)
- Processes Cannot Start Because of Incorrect Database Connection Configuration
- Virtual Host Mapping to Hide the ptg/rm URL
- Granting Change Password Privilege
- Password Policy Management
- Broken Images

16.5.1 Deploying XHTML+XForms, XHTML MP, and SimpleResult Documents

This section describes the following:

- Problem 1: OracleAS Wireless Does Not Transform XHTML +XFORM Documents Saved as .xhtml
- Problem 2: OracleAS Wireless Does Not Transform OracleAS Wireless Documents Generated Using PL/SQL
- Problem 3: Error When Generating XHTML with JSP
- Problem 4: Setting the MIME Type
- Problem 5: Header Information in PL/SQL Packages
- Problem 6: Errors for Valid XHTML Documents

Problem 1: OracleAS Wireless Does Not Transform XHTML +XFORM Documents Saved as .xhtml

I am trying to deploy an XHTML+XFORMs document on OracleAS Wireless. If I deploy the document as Mydoc.xhtml, OracleAS Wireless fetches this document but does not transform it. However, OracleAS Wireless transforms this document if I save `Mydoc.xhtml` as `Mydoc.jsp`.

Solution 1

Ensure that you set the appropriate MIME TYPE for the Web server. For example, for OC4J as a standalone Web server, you must configure

`ORACLE_HOME\wireless\j2ee\config\mime.types`:

`application/vnd.oracle.xhtml+xforms xhtml`

The Web server assigns the correct MIME TYPE (`vnd.oracle.xhtml+xforms`) to files with the `.xhtml` extension.

> **Note:** If you do the above mapping in the mime.types file, then all files with the `.xhtml` extension are served with the `application/vnd.oracle.xhtml+xforms` mime type. If you have other `.xhtml` documents in the application, choose a different extension (`.xforms`), for XHTML+XForms documents.

Problem 2: OracleAS Wireless Does Not Transform OracleAS Wireless Documents Generated Using PL/SQL

I am using PL/SQL to generate my OracleAS Wireless documents, but the server does not transform these documents. What should I do?

Solution 2

Verify that you set the correct MIME TYPE for the generated documents. For SimpleResult documents, you must set the MIME type as `text/vnd.oracle.mobilexml`. Example 16–4 describes how to set the MIME TYPE from PL/SQL using the `owa_util` package.

Example 16–4 Setting the MIME TYPE Using owa_util

```
owa_util.mime_header ('text/vnd.oracle.mobilexml', false, 'UTF-8')
owa_util.http_header_close;
htp.print ( '<?xml version="1.0" encoding="UTF-8" standalone="yes"?>' );
htp.print ( '<!DOCTYPE SimpleResult PUBLIC "-//ORACLE//DTD SimpleResult 1.1//EN"
"http://xmlns.oracle.com/ias/dtds/SimpleResult_1_1_0.dtd">' );
htp.print ( '<SimpleResult>' );
htp.print ( .....);
htp.print ( .....);
htp.print ( '</SimpleResult>' );
```

Problem 3: Error When Generating XHTML with JSP

I get the following error regarding my XHTML documents generated using JSP technology:

```
oracle.panama.adapter.AdapterException:
&lt;Line 5, Column 6>: XML-0109: (Fatal Error) PI with the name 'xml' can occur
only in the beginning of the document
```

How do I fix this?

Solution 3

Place the <?xml...?> processing instruction on the first line of the JSP (before the page directive) as illustrated in Example 16–5.

Example 16–5 Entering the Processing Instructions in a JSP

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<%@ page contentType="text/vnd.oracle.mobilexml; charset=UTF-8"%>
```

Problem 4: Setting the MIME Type

How do I set the MIME TYPE in the output in Oracle Application Server Wireless Release 904 (since it is more stringent than in Release 902)?

Solution

For `SimpleResult`, the content type must be set to `text/vnd.oracle.mobilexml` and the document must have a `DOCTYPE` declaration such as

```
<!DOCTYPE SimpleResult PUBIC "-//ORACLE//DTD SimpleResult 1.1//EN"
"http://xmlns.oracle.com/ias/dtds/SimpleResult_1_1_0.dtd">
```

Problem 5: Header Information in PL/SQL Packages

My JSPs, which work, have the header:

```
<?xml version = "1.0" encoding = "UTF-8" standalone="yes" ?>
<%@ page contentType="text/vnd.oracle.mobilexml; charset=UTF-8" %>
```

However, if I use a PL/SQL package rather than a JSP, how should I code this into the package body?

Solution 5

Because you are not using JSPs, there is no JSP engine to interpret and execute the `<jsp:...>` tags (the PL/SQL HTP package does not have such functionality). In fact, when you use a JSP page directive to set the content type, you are actually signaling the Web server to send a *content-type* response header back to the client (which, in this case, is the OracleAS Wireless Server). This is in contrast with the `<?xml ...?>` prolog and the `<!DOCTYPE`...> tag, both of which are part of the response body.

Note: The HTP.PRINT procedure only allows you to add content to the response body, not set response headers. Refer to the HTP package documentation for another procedure (such as `HTP.SET_CONTENT_TYPE`) that allows you to set the content type. If no alternative procedure will suffice, then adjust `mod_plsql`.

Setting the MIME Type Using the owa_util.mime_header

Using the owa_util.mime_header generates the correct SimpleResult as a text/vnd.oracle.mobilexml MIME type from PL/SQL:

```
owa_util.mime_header('text/vnd.oracle.mobilexml', true, 'UTF-8');
owa_util.http_header_close;
htp.print ('<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>');
htp.print ('<!DOCTYPE SimpleResult PUBLIC "-//ORACLE//DTD SimpleResult 1.1//EN"
"http://xmlns.oracle.com/ias/dtds/SimpleResult_1_1_0.dtd">');
```

Note: Setting *true* in the `owa_util.mime_header`

`('text/vnd.oracle.mobilexml', true, 'UTF-8')`;

inserts blank line before the line containing `<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>`. This creates an error response in OracleAS Wireless.

To avoid this error, set false in the `owa_util.mime_header`

`('text/vnd.oracle.mobilexml', false, 'UTF-8')` to prevent the insertion of the blank line, leaving `<?xml version...?>` on the first line.

Problem 6: Errors for Valid XHTML Documents

I get the following error even though the XHTML document is valid:

```
oracle.panama.adapter.AdapterException:
&lt;Line 5, Column 6>: XML-0109: (Fatal Error) PI with the name 'xml' can occur
only in the beginning of the document
```

Solution 6

You must set the content type of the XHTML document. You must put `<?xml ...?>` in the first line, as illustrated in Example 16–6.

Example 16–6 Setting the Content Type of an XHTML Document

```
<?xml version = "1.0" encoding = "UTF-8" standalone="yes" ?>
<% response.setContentType ("application/vnd.oracle.xhtml+xforms"); %>
```

16.5.2 TON and NPI Parameters

Problem

How do I configure TON (type of number) and NPI (national plan indicator) for SMS from Enterprise Manager?

Solution

You define the TON and NPI configuration parameters when you create the SMPP (Short Message Peer to Peer protocol) driver instance. These parameters are used to configure the SMSC (Short Message Service Center). Because the connection configuration of the (SMPP) driver must concur with the SMSC configuration, you should consult with the SMSC administrator for the correct values for these parameters.

16.5.3 Configuring SMPP Parameters

Problem

What are the configuration parameters for the SMPP driver?

Solution

Table 16–2 describes a list of the parameters, with some example values set.

Table 16–2 Parameters of the SMPP Driver

Parameter	Description	Example Value
`sms.account.id`	The ID of the account or the short number.	200
`sms.smpp.server.host`	The host name or IP address of the SMSC server.	192.168.10.10.
`sms.smpp.transmitter.system.id`	The system ID of the transmitter to the SMPP server.	oracle
`sms.smpp.transmitter.system.type`	The type of system ID of the transmitter to the SMPP server.	CERT
`sms.smpp.transmitter.system.password`	The password for the transmitter of the SMSC.	oracle78
`sms.server.transmitter.port`	The number of the SMSC server's listening port for the transmitter.	5135
`sms.smpp.receiver.system.id`	The system ID of the receiver to the SMSC server.	oracle
`sms.smpp.receiver.system.type`	The tyoe of system ID of the receiver of the SMPP server.	CERT
`sms.smpp.receiver.system.password`	The password for the receiver to the SMSC server.	oracle78
`sms.server.receiver.port`	The number of the SMSC listening port for the receiver.	5136
`sms.server.default.encoding`	The default encoding for text messages.	IA5
`sms.local-sending.port`	The local port used by the SMPP driver to make the out-going, sending connection.	
`sms.local-receiving.port`	The local port used by the SMPP driver to make an outgoing, receiving connection.	
`sms.local.address`	The local hostname or IP address of the server that runs the SMPP driver.	
`sms.server.source.ton`	Type of number of the source address (the account ID).	00
`sms.server.source.npi`	Nation Plan Indicator of the source address.	01
`sms.server.destination.npi`	National Plan Indicator of the destination address.	00
`sms.server.destination.ton`	Type of number of the destination address.	00
`sms.window.size`	The windowing size.	1
`sms.bulk.sending`	Whether to use the submit_multi command for sending messages to a group of subscribers (if possible).	false
`sms.payload.sending`	Whether to use the message_ payload field for sending when the short_message field is okay.	true

Table 16–2 (Cont.) Parameters of the SMPP Driver

Parameter	Description	Example Value
`sms.message.maxchunks`	The maximum chunks that a long message can be split into. A negative value means no limitation.	-1
`sms.message.chunksize`	The maximum number of characters a chunk can have.	160
`sms.enquire-link.interval`	The interval, in seconds, after which the enquire link is called. This feature is disabled if the interval is less than 0	15
`sms.throttling.delay`	The delay, in seconds, after the sending restarts after the throttling error is received from the SMSC. This feature is disabled if the delay is less than 0.	30
`sms.extra.error-code`	The extra error codes that could be returned by the SMSC and requires sending the message again.	
`sms.bind.retry.delay`	The time, in seconds, that ESME (Extra Short Message Entity) waits for an enquiry link response before retrying bind to SMSC.This feature is disabled if this parameter or the sms.enquire-link.interval is less than 0	30
`sms.registered.delivery.mark`	The mark to disable a registered delivery flag if the SMSC does not support registered delivery flag functions.	0xFF
`sms.wireless.network`	The type of network used to send SMS.	GSM.
`sms.priority.allowed`	Designates the highest priority allowed for a message.	0

16.5.4 Using Third-Party Authenticators with OracleAS Wireless

Problem

How do I plug a third-party authenticator into OracleAS Wireless? Can I do this using Oracle SSO (Single Sign-On)?

Solution

You can integrate third-party SSO solutions with Oracle SSO. You should consider integration rather than removing Oracle SSO.

For non-OracleAS Wireless applications, integration is fairly straightforward. See *OracleAS Single Sign-On Integration with Third-Party Single Sign-On Products* on the Oracle Technology Network (http://www.oracle.com/technology/), specifically for RSA ClearTrust. See RSA Security's documentation regarding SSO at the RSA Security Web site (www.rsasecurity.com).

16.5.5 Problem Resetting Passwords

Problem

When I installed OracleAS Wireless on a Solaris server it worked correctly. Now, however, the *oidadmin* password expired. I tried to change the password using OID, but when I try to log into OID as the *ias_admin* user, I get the error *Password policy error, your account is locked*. What should I do?

Solution

You received the error because of the password expiry time for realms, which expires users' passwords after a certain time. To log in as the *ias_admin* user, you must increase the password expiry time value as follows:

1. Open the oidadmin tool on the infrastructure-tier machine. It is in:

   `ORACLE_HOME/bin`.
2. Log on as *cn=orcladmin*
3. Go to *Password Policy for Realms cn=....* inside the Password Policy Manager. Increase the value for the *Password Expiry Time* field. This field value is represented in seconds, so you must enter a very large number to avoid password problems in the near future.
4. Click **Apply**. You can now use *orcladmin* with the old password.

16.5.6 Postinstallation Configuration

Problem

I installed OracleAS Wireless without running the OracleAS Wireless configuration assistant. Then I ran `wirelessCA.sh` from the command line without problems (it returned *success*), but not all of the components are installed. How can I fix this?

Solution

Perform the following on the middle tier to configure OracleAS Wireless as a post-installation step:

1. Using the Oracle Enterprise Manager 10*g* Application Server Control, first selecting the OracleAS Wireless middle-tier node on the *Farm* page of the Oracle Enterprise Manager 10*g* Application Server Control. The *Home* page appears.
2. Under *System Components*, click **Configure Component**. The *Select Component* page displays.
3. Select *Wireless* from the list.
4. Click **Continue**.
5. Enter the password for the OID super user (*orcladmin*).
6. Click **Finish**.
7. Wait 10 minutes to deploy OC4J_Wireless and complete the process.
8. Start the OracleAS Wireless Server from EM by selecting *Wireless* and clicking **Start** (under *System Components*, or use the OPMN command set options).

16.5.7 Error Enqueuing Received Messages

Problem

I configured a driver to receive messages. When I tried to send messages to the driver, I received a `failed to enqueue a received message` error in the log.xml file. Is it a bug?

Solution

This is not a bug. You received the error because the Messaging Server does not know where to dispatch the received message; there must be a registered application listening for incoming messages from the driver. This application must have the addressing information for the driver (such as the incoming e-mail address for an e-mail driver).

> **Note:** If you configure a driver only to see if it can receive messages, you can ignore such errors.

If you configure the driver so that an application can receive messages received by the driver, you must ensure that the application is running (or at least has run once before). If the application is running (or has run at least once before), it is possible that the application did not register the correct endpoint to the transport layer. An access point (also known as an endpoint) is a pair (a delivery type supported by the driver and the incoming address for which the driver listens). The application should be named the *Messenger* instance. Call the `addEndPoint(EndPoint ep)` method of Messenger to register the endpoint, then call the `start()` method of the Messenger instance.

If you configured the driver to work with Async Listener, the error indicates that Async Listener is not properly configured, is not running, or has not run at least once after you properly configured it. The Async Listener is a built-in application to listen for received messages. The access points to which the Async Server listens can be configured with System Manager. See Section 3.10.2.1 on how to configure Async Server listening access points.

When configuring the Async Listener access points, make sure you correctly entered the incoming address to which the driver listens. This address must be one of the Async Listener's access points.

Ensure the Async Listener is running (you can check this status of the Async Listener from the *Home* page of the System Manger. For more information on the Async Listener and other Web-based applications, see Section 3.10).

In the *System Logging* page (see Section 3.4.2), select the NOTIFY logging level, then start the *OC4J_Wireless* component in the *Home* page of the Oracle Enterprise Manager to ensure that the Async Listener is running. If the Async Listener is running, then the `log.xml` file should contain a message such as *Async Listener started*.

16.5.8 Errors in Receiving Messages

Problem

I sent a message but it has not been received. Where is it?

Solution

To best answer this question, the implementation details should be examined, but without going that far, here are possibilities:

- Since the Messaging Server depends on each driver to actually deliver the message, the message could be in the external messaging gateway, or still in the Messaging Server persistence store. If it stays in the Messaging Server store, it is not delivered to the external messaging gateway. This could happen because a proper Messaging Server is not up, or that the message is routed to a driver that does not have a configured driver instance. In this case, the sending status is: *The message was accepted*.
- If the message has been delivered to the external messaging gateway, then the status message *The message has been delivered to the messaging gateway successfully* appears after the message is delivered to the external messaging gateway. Check the external messaging gateway for the status of the message.

16.5.9 Setting the Proxy in XMS

Problem

I use an HTTP proxy to connect to t he Internet. How do I set the proxy in XMS?

Solution

An XMS client connects to the XMS Web service through the HTTP protocol. As such, a proxy must be set if one is needed to communicate over HTTP from the host that the XMS client runs on to the host the XMS Web service runs on. Set the proxy programmatically using the XMS API as follows:

```
XMSSender.setProxy(host,port)
```

or

```
XMSSimpleSender.setProxy(host,port)
```

16.5.10 Unable to Reach XMS Gateway

Problem

Why do I receive a *No response from gateway* message when I send a message using the XMS API to a messaging gateway.

Solution

You received this message either because the messaging gateway is not running or because it cannot be reached. To reach the messaging gateway from the client machine:

- Use the command: `telnet <gateway URL> <port>` to check if the client machine can contact the messaging gateway. For example, enter the following command:

  ```
  telnet messenger.oracle.com 80
  ```

  If the Telnet hangs, it is likely that the outgoing network traffic is blocked by a firewall.
- Otherwise, use the following command in the Telnet window:

  ```
  GET /xms/webservices <Enter <Enter
  ```

You should receive a short reply from the gateway such as:

SOAP Server

Sorry, I don't speak via HTTP GET- you have to use HTTP POST to talk to me. (The actual replies may vary.) If the client machine requires a proxy to access the Internet, be sure to set the proxy settings correctly.

16.5.11 Dialing Country Codes with SMS or MMS

Problem

In my country, I must add a 00 or 011 prefix before the country code when I make international long distance calls. Must I do the same when I am sending out SMS or MMS?

Solution

No. Do not add those prefixes. You need only provide the country code, and cell phone number with area code (if any). However, note that you must provide the prefix for voice notifications.

16.5.12 Bad Username/Password Exceptions

Problem

I am getting the exception: *Bad username or password* when I contact the messaging gateway. What is wrong?

Solution

Possible reasons include:

- The password is not correct.
- The username does not exist on that gateway.
- The user is not enabled.
- The user does not have quota available.

Verify that the user name and password are valid. If they are valid and this error occurs when accessing the Oracle-hosted messaging instance (`http://messenger.oracle.com/xms/webservices`), contact `mobilesupport_ww@oracle.com` to determine the exact problem.

16.5.13 Error When Checking Message Status

Problem

I received the message: *Only sender who is a registered user is allowed to check the message status* when I check message status using the `getStatus()` method in the XMS API. What is wrong?

Solution

Possible reasons include:

- The user name is not registered on that gateway. Some gateways allow anonymous users to access free trials before registering. They can send out some messages, but they may not check the sending status of those messages.

- The user is not the sender of the message.

16.5.14 AddressData Class and Message Sending Failure

Problem

I cannot send a message using the code:

```
AddressData toAddress = new AddressData("a.user@oracle.com", "email");
Packet packet = new Packet();
packet.addRecipient(toAddress);
```

I notice that the `workOrders` array returned by the `sender.sendMsg()` method is empty. What should I do?

Solution

The `AddressData` class should not be constructed directly (it is declared public for technical reasons, but should not be constructed). Although the JavaDoc explicitly notes that you should not construct this class, some users do anyway. Rather than construct the `AddressData class`, use the AddressDataFactory instead (as illustrated in Example 16–7).

Example 16–7 AddressDataFactory

```
AddressData toAddress =
AddressDataFactory.getInstance().createAddressData("email:a.user@oracle.com")
```

16.5.15 Unable to Get Delivery Status Callback

Problem

I cannot get delivery status callback from the messaging gateway. What should I do?

Solution

Check the following:

- Is the Status Listener set?

  The Status Listener must be set before sending out the message. In Example 16–8, the client requests that the messaging gateway sends a status update to `http://my-pc.foo.bar:3900`. The `MyStatusListener` class is notified when new a status arrives.

Example 16–8 Client Requesting a Status Update

```
MyStatusListener myListener = new MyStatusListener();
xmsSender.setStatusListener(myListener, "my-pc.foo.bar", 3900);
```

- Is the tracking flag set?

  The tracking flag must be explicitly set to get callback. For example:

  ```
  MessageInfo mInfo = new MessageInfo();
  mInfo.setTracking(MessageInfo.TIGHT_FULL_TRACK);
  pkt.setMessageInfo(mInfo);
  ```

- Can the client machine be reached by messaging gateway?

The messaging gateway server must be able to reach the specified host name. Firewalls must be configured to allow HTTP requests from messaging gateway. The client machine may be placed in DMZ if the gateway is on the Internet.

16.5.16 Slow Performance After Deploying Applications

Problem

The OC4J_Wireless instance became slow or non-responsive after I deployed my applications. How can I address the problem?

Solution

The built-in applications in the OC4J_Wireless container are tuned to support a large number of concurrent requests. However, custom applications may introduce slowdowns and cause poor performance. Custom applications include HTTP adapter services as well as custom implementations of hook interfaces (such as folder renderer, authentication, authorization, and pre- and post- processor).

You can identify performance bottlenecks in the OC4J_Wireless container by analyzing the thread dumps generated while the system is under stress tests. On UNIX and LINUX operating systems, you can send `SIGQUIT`, using the command `(kill -3 PID)` to the OC4J_Wireless process to generate the thread dumps in the standard output. The standard output stream for OC4J_Wireless is piped to the log file in the `ORACLE_HOME/opmn/logs` directory.

You must identify the *PID* (process ID) of the OC4J_Wireless process. The process ID can be identified in the output of the UNIX command:`/usr/ucb/ps -auxww | grep OC4J_Wireless`.

Take two or more thread dumps at 30 second intervals while running stress test scenarios. Design test scenarios so that thread dumps will capture the snapshots of the applications at several stages of their request processing life cycles. If many threads are waiting at the same location in the request processing life cycle, that location constitutes a bottleneck in the applications. This may be due to several factors; here are two common ones:

- Resource contention (such as contention for JDBC connections)—if the problem is due to resource contention, increase the number of resources, but first make sure that there are no resource leaks in the application. JDBC connection leaks are a common cause of resource contention problems.
- Sizes of critical sections—putting time-consuming operations inside the critical sections. Reduce the size of critical sections to reduce synchronization among threads. In many applications, it is fairly easy to identify the segments of the critical sections that can be safely moved outside the critical sections. Tune the applications to reduce the number of threads held up. You may have to repeat several iterations of tuning (analyze thread dumps, modify code, restart OC4J_Wireless and test scenarios, and dump threads) until major bottlenecks are eliminated. You should see a noticeable improvement in the performance of the applications each time you eliminate a major bottleneck.

16.5.17 Finding Oracle Sensor Edge Server Information and Downloads

Problem

I need additional information about the Oracle Sensor Edge Server. Where can I find more information and access the downloads? Is there a Web site specific to the Oracle Sensor Edge Server?

Solution

Oracle Technology Network (http://www.oracle.com/technology/) provides the lastest filters, drivers, and the Edge Development Kit for developing extensions to the Oracle Sensor Edge Server, as well as information and tutorials.

16.5.18 Intermittent Browser Crashes

Problem

An intermittent problem has been reported in which the Microsoft Internet Explorer 6.x running on the Microsoft Windows XP operating system crashes while a user is using the Wireless Tools. The problem typically occurs on resource-limited servers.

Solution

The problem has not been reported when Netscape or Mozilla browsers are used, so a solution to the problem is to use one of these browsers.

16.5.19 Incorrect Display of Non-ASCII Characters on Some Devices

Problem

When I try to display non-ASCII characters on my device, I sometimes get unreadable characters. How can I display non-ASCII characters?

Solution

There may be several reasons for this problem. Verify the following:

- Make sure that the original content (the XHTML or Mobile XML document) contains the correct characters. Very often editors (such as vi, Notepad, and others) corrupt the file content if it contains non-ASCII characters. To make sure that the original content is correct, try to download the content using a PC browser, and verify that the non-ASCII characters are correct. It may be best to use JSPs or Servlets for the content and escape the non-ASCII characters using the `Java \uxxxx` syntax, where `xxxx` is the appropriate hexadecimal value of the character.

  For more details about Unicode Escapes, refer to the Unicode Escapes of the Java Language Specification at:

  http://java.sun.com

- Make sure that the HTTP response contains the correct HTTP headers; specifically, make sure that the Content-Type header contains the correct encoding in the charset. By default, the HTTP protocol assumes that the content is encoded using ISO-8859-1 encoding. It is best to use JSPs or Servlets for the content. It is best to use JSPs or Servlets for the content and include the lines described in Example 16–9 within the JSP.

Example 16–9   Charset Encoding in the Content-Type HTTP Header

```
<?xml version = "1.0" encoding = "UTF-8" standalone="yes" ?>
<%@ page contentType="text/vnd.oracle.mobilexml; charset=UTF-8" %>
<%
request.setCharacterEncoding("UTF-8");
%>
```

In this case the content is MobileXML but the same applies to XHTML. Example 16–10 illustrates simple MobileXML:

Example 16–10   MobileXML

```
MobileXML
<?xml version = "1.0" encoding = "UTF-8" standalone="yes" ?>
<!DOCTYPE SimpleResult PUBLIC "-//ORACLE//DTD SimpleResult 1.1.0//EN"
"http://xmlns.oracle.com/ias/dtds/SimpleResult_1_1_0.dtd">
<%--=======================================================
| Copyright (c) 1999-2004 Oracle Corporation, Redwood Shores, CA, USA
| All rights reserved.
+=======================================================--%>
<%@ page contentType="text/vnd.oracle.mobilexml; charset=UTF-8" %>
<%@ page language="java" %>
<%@ page import="java.util.*" %>
<%@ page import="java.text.*" %>
<%
request.setCharacterEncoding("UTF-8");
%>
<%-- Prevent Page Caching --%>
<%
response.setHeader("Cache-Control", "no-store"); // HTTP 1.1
response.setHeader("Pragma", "no-cache"); // HTTP 1.0
response.setHeader("Expires", "0"); // prevents caching at the proxy server
%>
<%
Date date = new Date();
DateFormat df = DateFormat.getDateTimeInstance(DateFormat.FULL, DateFormat.FULL);
String time = df.format(date);
%>
<SimpleResult>
<SimpleContainer>
<SimpleText>
<SimpleTitle>OracleAS Wireless</SimpleTitle>
<SimpleTextItem>Hello World!</SimpleTextItem>
<SimpleTextItem><%=time%></SimpleTextItem>
</SimpleText>
</SimpleContainer>
</SimpleResult>
```

- If you cannot ensure that the correct Content-Type HTTP header is sent by the content source, then you can configure the OracleAS Wireless server to always use a predefined character encoding for this service. To do this, you must set a value for the `INPUT_ENCODING` input parameter of the service. The disadvantage of this approach is that the character encoding is hard-coded in the OracleAS Wireless Server and the content source cannot change it. In addition, this parameter is only available in the device portal, not in the Multi-Channel Server.
- Make sure that the `Accepted Character Encodings` attribute of the Device object contains the correct character encoding. The OracleAS Wireless attempts to get the character encoding of the device from the Accept-Charset HTTP header,

but if that header is not sent by the device, the OracleAS Wireless uses the value of the `Accepted Character Encodings` attribute instead.

16.5.20 Service Error Caused by HTTPAdapter Error

Problem

I get a Service Error on my device, and the log file seems to indicate that the problem involves the HTTPAdapter. How can I be sure what the problem is, and how can I fix it?

Solution

When you get a Service Error on the device, read the error message in `ORACLE_HOME/wireless/log.xml`(Example 16–11). When the error is caused in the HTTPAdapter, then you will see an error message such as the one below (the most import parts of the message are italicized and in bold):

The segments of the error message on which you should focus are:

- The message in `oracle.panama.adapter.AdapterException` which gives general information about what may have gone wrong. Problems are usually related to HTTP protocol problems or XML parser errors.
- For HTTP protocol problems, read the `HTTP response status` and `HTTP response message`. It may be useful to get the URL from the Original URL and Effective URL (those are the absolute URLs to content providers), and try to access them directly from a PC browser. If you use JSP and there is a problem with the JSP, then you will get an *HTTP response status 500*. When you access the JSP directly from the PC browser then you will see the exact error message.
- For an XML parser error, read the `Response`. It contains the response from the content provider. Please note that the standard XML entities are escaped. To help you identify the problem, take the response, save at in a separate file and then validate the XML.

Example 16–11 The Log.xml Error Message

```
<MESSAGE>
  <HEADER>
    <TSTZ_ORIGINATING>2004-06-08T13:48:52.818-07:00</TSTZ_ORIGINATING>
    <ORG_ID>ORACLE</ORG_ID>
    <COMPONENT_ID>WIRELESS</COMPONENT_ID>
     <MSG_TYPE TYPE="ERROR"></MSG_TYPE>
    <MSG_LEVEL>1</MSG_LEVEL>
    <HOST_ID>as-host</HOST_ID>
    <HOST_NWADDR>111.222.333.444</HOST_NWADDR>
    <THREAD_ID>AJPRequestHandler-ApplicationServerThread-6</THREAD_ID>
    <USER_ID>vu_1086727714427_pA39cnouBrKQ45A1</USER_ID>
  </HEADER>
  <CORRELATION_DATA>
    <EXEC_CONTEXT_ID><UNIQUE_ID>1086727732:111.222.333.444:2580:3348:8</UNIQUE_ID>
    <SEQ>0</SEQ></EXEC_CONTEXT_ID>
  </CORRELATION_DATA>
  <PAYLOAD>
    <MSG_TEXT>
null
oracle.panama.PanamaException
oracle.panama.adapter.AdapterException:
The resource at http://as-host:7777/mcs/examples/index.js
```

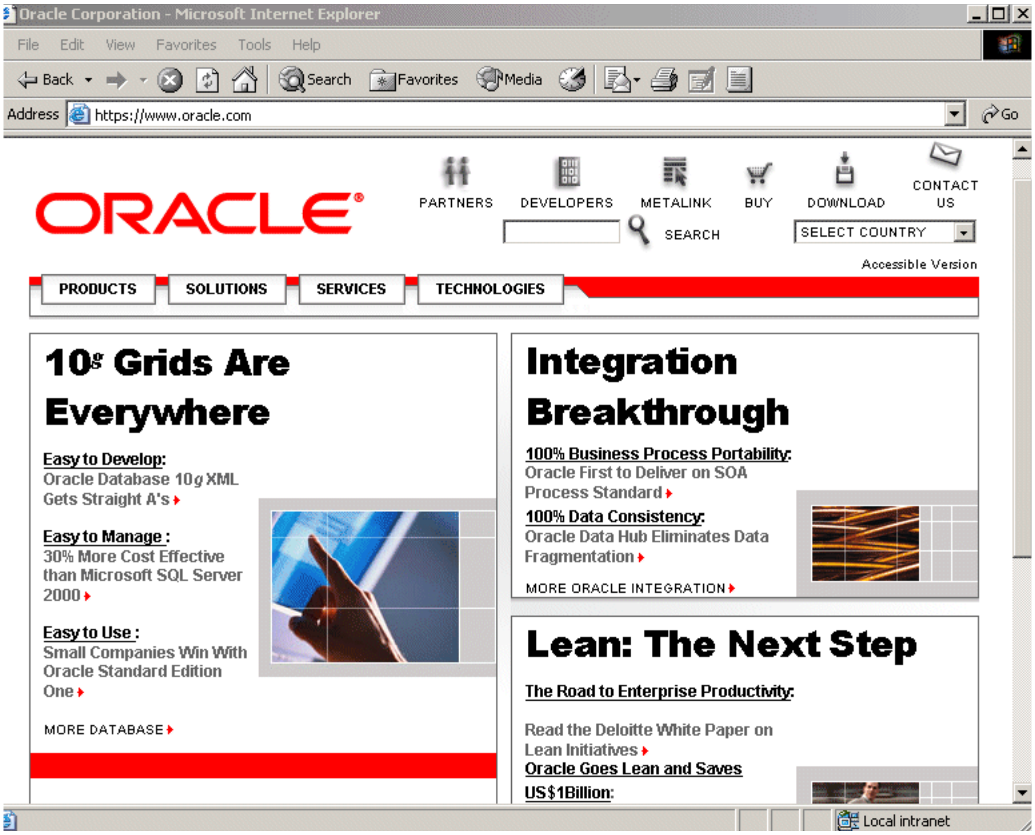
```
could not be loaded from content provider.HTTP(S) Error: 404 : Not Found
Original URL: "http://as-host:7777/mcs/examples/index.js"
Effective URL: "http://as-host:7777/mcs/examples/index.js"
Query String: ""
Request HTTP headers:
x-oracle-device.secure: false
x-oracle-user.name: vu_1086727714427_pA39cnouBrKQ45A1
accept-language: en-US
x-oracle-device.orientation: landscape
accept: application/vnd.oracle.xhtml+xforms,
text/vnd.oracle.mobilexml, application/vnd.wap.xhtml+xml,
application/xhtml+xml;profile="http://xmlns.oracle.com/ias/dtds/xhtml+xforms",
application/xhtml+xml;profile="http://www.wapforum.org/xhtml",
application/xhtml+xml,
application/xml, text/xml, application/vnd.oracle.xad, */*
x-oracle-device.class: pcbrowser
x-oracle-user.deviceaddress: 111.222.333.444
user-agent: PTG/2.0 (Oracle9iAS Wireless 9.0.4.0; screen; color8; 1024x800; tables)
x-oracle-user.authkind: unauthenticated
x-oracle-user.locale: en-US
x-oracle-device.maxdocsize: 0
oracle-ecid: 1086727732:111.222.333.444:2580:3348:8,0
x-oracle-orig-accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword,
application/x-shockwave-flash, */*
x-oracle-orig-user-agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
(R1 1.3); .NET CLR 1.1.4322)
x-oracle-user.deviceid: vu_1086727714427_pA39cnouBrKQ45A1
x-oracle-device.name: HTML40
x-oracle-service.parent.url: /ptg/rm?PAsid=pA39cnouBrKQ45A1&amp;PAgoHome=1
x-oracle-wireless.https.url: https://as-host:4443/ptg/rm
x-oracle-home.url: /ptg/rm?PAsid=pA39cnouBrKQ45A1&amp;PAgoHome=1
x-oracle-user.userkind: virtual
x-oracle-wireless.http.url: http://as-host:7777/ptg/rm
x-oracle-service.home.url: /ptg/rm?PAsid=pA39cnouBrKQ45A1&amp;PAoid=225
HTTP response status: 404
HTTP response message: Not Found
Response HTTP headers:
x-pad: avoid browser bug
date: Tue, 08 Jun 2004 20:48:52 GMT
connection: Keep-Alive
content-type: text/html
content-length: 145
keep-alive: timeout=15, max=100
server: Oracle-Application-Server-10g/10.1.2.0.0 Oracle-HTTP-Server
cache-control: private
Response:
&lt;HTML>&lt;HEAD>&lt;TITLE>404 Not Found&lt;/TITLE>&lt;/HEAD>&lt;BODY>&lt;H1>
404 Not Found&lt;/H1>
Resource /mcs/examples/index.js not found on this server&lt;/BODY>&lt;/HTML>
at oracle.panama.adapter.http.HttpAdapter.fetchData(HttpAdapter.java:829)
at oracle.panama.adapter.http.HttpAdapter.fetchResult(HttpAdapter.java:578)
at oracle.panama.adapter.http.HttpAdapter.invoke(HttpAdapter.java:468)
at oracle.panama.core.MasterServiceImpl.invokeAdapter(MasterServiceImpl.java:386)
at oracle.panama.core.MasterServiceImpl.getPAElementInternal(MasterServiceImpl.java:552)
at oracle.panama.core.AliasImpl.getPAElementInternal(AliasImpl.java:344)
at oracle.panama.core.ServiceImpl.invoke(ServiceImpl.java:772)
at oracle.panama.rt.common.Controller.fetchContent(Controller.java:489)
at oracle.panama.rt.common.Controller.fetchContent(Controller.java:509)
at oracle.panama.rt.common.AbstractController.processContent(AbstractController.java:690)
at oracle.panama.rt.common.AbstractController.doExecute(AbstractController.java:660)
at oracle.panama.rt.common.ConnectionImpl.doExecute(ConnectionImpl.java:2043)
at oracle.panama.rt.common.ConnectionImpl.execute(ConnectionImpl.java:2140)
at oracle.panama.servlet.ParmImpl.doGet(ParmImpl.java:208)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:740)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
at com.evermind.server.http.ResourceFilterChain.doFilter(ResourceFilterChain.java:65)
at oracle.panama.servlet.MultipartFilter.doFilter(MultipartFilter.java:133)
at com.evermind.server.http.ServletRequestDispatcher.invoke(ServletRequestDispatcher.java:604)
at
com.evermind.server.http.ServletRequestDispatcher.forwardInternal(ServletRequestDispatcher.java:317
)
at com.evermind.server.http.HttpRequestHandler.processRequest(HttpRequestHandler.java:790)
at com.evermind.server.http.AJPRequestHandler.run(AJPRequestHandler.java:208)
at com.evermind.server.http.AJPRequestHandler.run(AJPRequestHandler.java:125)
at
com.evermind.util.ReleasableResourcePooledExecutor$MyWorker.run(ReleasableResourcePooledExecutor.ja
va:192)
at java.lang.Thread.run(Thread.java:534)
</MSG_TEXT>
</PAYLOAD>
```

16.5.21 Downloading SSL Root Certificates

Of the many ways to download SSL root certificates, one of the easiest methods is to use a desktop browser to download the certificate. The following steps use Internet Explorer.

1. Open Internet Explorer and access the HTTPS Web site containing the SSL root certificate that you want to download.

Figure 16–4 Web Page for Downloading SSL Root Certificates



Address https://www.oracle.com

10g Grids Are Everywhere

Easy to Develop: Oracle Database 10*g* XML Gets Straight A's

Easy to Manage : 30% More Cost Effective than Microsoft SQL Server 2000

Easy to Use : Small Companies Win With Oracle Standard Edition One

Integration Breakthrough

100% Business Process Portability: Oracle First to Deliver on SOA Process Standard

100% Data Consistency: Oracle Data Hub Eliminates Data Fragmentation

Lean: The Next Step

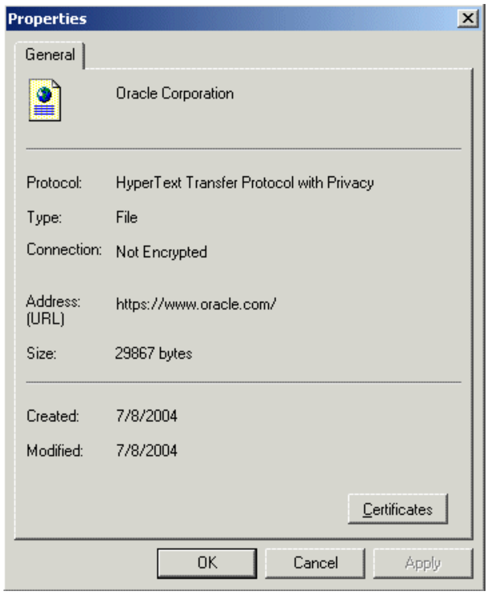
The Road to Enterprise Productivity:

Read the Deloitte White Paper on Lean Initiatives

Oracle Goes Lean and Saves US$1Billion:

2. On the page, right-click the mouse and select **Properties** from the drop-down menu. The *Properties* window appears (Figure 16–5).

Figure 16–5 The Properties Window



Oracle Corporation

Protocol: HyperText Transfer Protocol with Privacy

Type: File

Connection: Not Encrypted

Address: (URL) https://www.oracle.com/

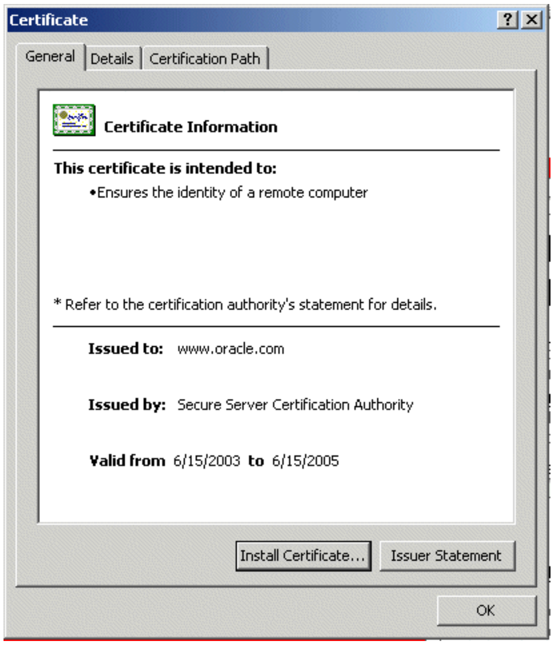
Size: 29867 bytes

Created: 7/8/2004

Modified: 7/8/2004

3. Click the **Certificates** button. The certificate information dialog appears (Figure 16–6).

Figure 16–6 The Certificates Information Dialog



Certificate

Certificate Information

This certificate is intended to:

- Ensures the identity of a remote computer

* Refer to the certification authority's statement for details.

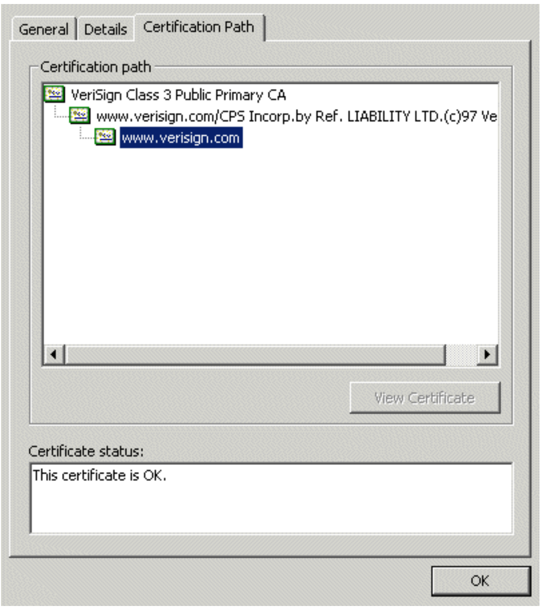
Issued to: www.oracle.com

Issued by: Secure Server Certification Authority

Valid from 6/15/2003 **to** 6/15/2005

4. Click the **Certification Path** tab. The *Certificate Path* dialog appears (Figure 16–7), listing the certifcates.

Figure 16–7 The Certificate Path Dialog



Certification path

- VeriSign Class 3 Public Primary CA
  - www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 Ve
    - www.verisign.com

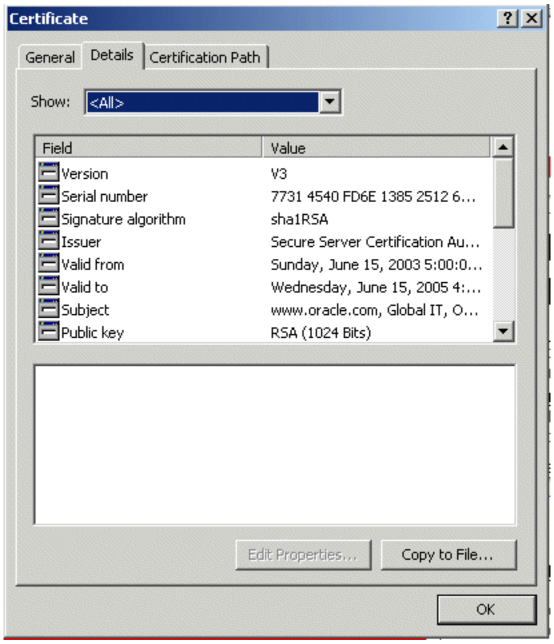
Certificate status:

This certificate is OK.

5. Select the first certificate in the *Certificate* path list.

6. Click the **Details** tab. The *Details* dialog appears (Figure 16–8).

Figure 16–8 The Details Tab of the Certificate Dialog



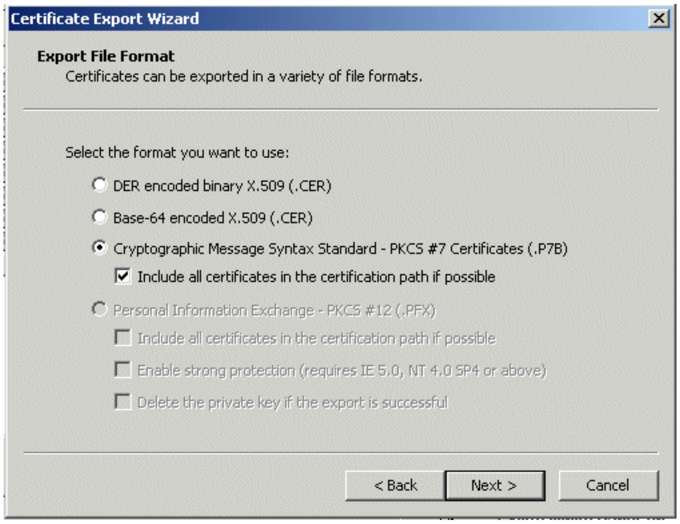
Show: <All>

Field	Value
Version	V3
Serial number	7731 4540 FD6E 1385 2512 6...
Signature algorithm	sha1RSA
Issuer	Secure Server Certification Au...
Valid from	Sunday, June 15, 2003 5:00:0...
Valid to	Wednesday, June 15, 2005 4:...
Subject	www.oracle.com, Global IT, O...
Public key	RSA (1024 Bits)

7. Click **Copy to File...** . The Certificate Export Wizard's *Welcome* screen appears.

8. Click **Next**. The *Export File Format* dialog appears (Figure 16–9).

Figure 16–9 The Export File Format Dialog



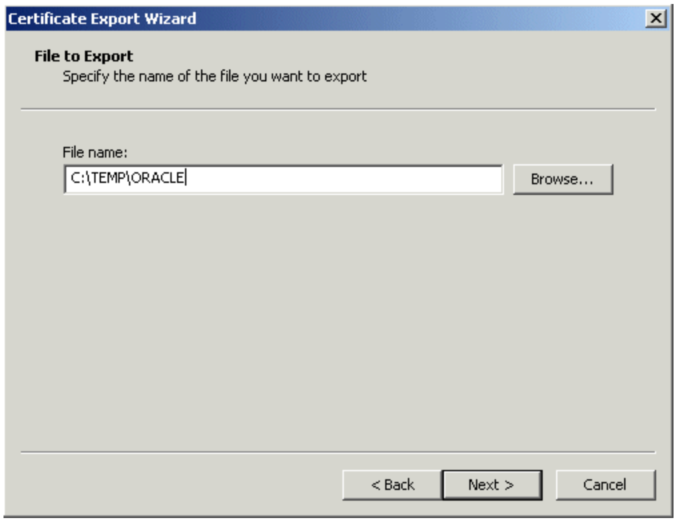
Export File Format
Certificates can be exported in a variety of file formats.

Select the format you want to use:

○ DER encoded binary X.509 (.CER)
○ Base-64 encoded X.509 (.CER)
● Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)
☑ Include all certificates in the certification path if possible
○ Personal Information Exchange - PKCS #12 (.PFX)
□ Include all certificates in the certification path if possible
□ Enable strong protection (requires IE 5.0, NT 4.0 SP4 or above)
□ Delete the private key if the export is successful

9. Select the PKCS #7 (or Base-64) format, then click **Next**. A dialog appears (Figure 16–10) requiring you to enter the file name and path of the file that you want to export.

Figure 16–10 Exporting the File



File to Export

Specify the name of the file you want to export

File name:

C:\TEMP\ORACLE

10. Specify a file name and location for the file in which certificate information will be saved. Do not specify the file extension; the wizard adds the extension automatically. Be sure to note the location of the file.
11. Click **Next**. A *Summary* page appears with the information that you supplied to the Wizard.
12. Click **Finish**. The SSL root certificate is stored in the file.
13. Select the next certificate in the *Certificate path* list and repeat Steps 6 through 12. You must download all of the available certificates.

16.5.22 Setting Up OracleAS Wireless to Support HTTPS

Problem

How do I set up OracleAS Wireless to support HTTPS?

Background

The OracleAS Wireless Server is a middle tier between mobile devices and mobile content. Some devices use the HTTP(S) protocol to communicate with the OracleAS Wireless Server. The OracleAS Wireless Server uses the HTTP(S) protocol to retrieve content from content providers. This means that there are two HTTP(S) hops between a mobile device and mobile content. They both can be configured to use HTTPS. To support HTTPS between a mobile device and the Wireless Server, you must configure the Oracle HTTP server to accept HTTPS requests. See the *Oracle HTTP Server Administrator's Guide* for more information.

Solution

For content providers, the OracleAS Wireless Server functions as a virtual browser. Like other browsers, the OracleAS Wireless Server uses certificates to positively identify certification authorities and publishers. Before you can use HTTPS to connect to a content provider, you must download the content provider's SSL root certificates (refer to Section 16.5.21 for more information). After you download the certificate, copy it to the OracleAS middle-tier machine.

Note: If you have more than one middle tier, then you should copy the file to all middle- tier machines. The file must be in the same location on all of the middle-tier machines.

Once you have copied the certificate file to the middle-tier machine, you must configure the OracleAS Wireless Server to use that certificate. If you have more than one middle-tier machine, you must configure only the first middle tier; all middle tiers will use the same settings (that is why the certificate file must be in the same location on all middle-tier machines).

To set a certificate for the OracleAS Wireless Server:

1. From the Oracle Enterprise Manager 10*g* Application Server Control, select the OracleAS Wireless middle-tier node on the *Farm* page of the Oracle Enterprise Manager 10g Application Server Control and then click **Wireless** on the *Application Server Home* page to access the OracleAS Wireless system management functions described in Chapter 3, "Managing the OracleAS Wireless Server".
2. Click the *Site Administration* tab of the OracleAS Wireless *Home* page. From the *General Configuration* section, select **HTTP, HTTPS Configuration**. The *HTTP, HTTPS Configuration* screen appears.
3. The SSL section enables you to add a certificate. The OracleAS Wireless Server supports two certificate file formats: *Base64* or *PKCS#7*.
4. Enter the absolute path to the certificate file. You can set more than one certificate by clicking **Add Another Row**.

Note: If you cannot copy the SSL certificate files to the same location on all of the middle-tier machines, then you must add the absolute path to the same file on all of the middle-tier machines as a different file. Because you are adding the same file multiple times, you may receive error messages in `$ORACLE_HOME/wireless/logs/log.xml` stating that the OracleAS Wireless Server could not find some of the certificates. You can ignore this error if you verify that the absolute path is entered correctly for all of the certificate files.

---

5. Click **OK**. For more information, see Section 3.9.1.1.3. and Section 10.4.2.

16.5.23 Out of Memory Exceptions in Log Files

Problem

I am receiving *Out of Memory* exceptions in my log file; how do I correct this?

Solution

When *Out-of-Memory* exceptions appear in either the `ORACLE_HOME/wireless/logs/log.xml` or the OPMN log files, increase the OC4J_Wireless JVM memory size:

1. From the Oracle Enterprise Manager 10*g* Application Server Control, first select an OracleAS Wireless middle-tier node on the *Farm* page of the Oracle Enterprise Manager 10*g* Application Server Control and then select the **OC4J_Wireless** process on the *Application Server Home* page and then select **Administration**.

2. Select **Server Properties** (located under *Instance Properties*). The *Server Properties* page for the OC4J_Wireless Server appears.
3. In the *Java Options* field, enter *-Xms128 -Xmx256*. This sets the JVM Heap Memory to be started at 128MB and could expand to 256MB.
4. Click **Apply**.
5. Shut down and restart OC4J_Wireless; the JVM will be started with the new memory configuration data. For more information on JVM tuning, see Section 13.9.2.

16.5.24 Non-ASCII User Name Corrupted in HDML Browser

Problem

A user has reported that corrupted user name data displayed in their HDML browser. Are non-ASCII data not supported?

Solution

The problem occurs not because of non-ASCII data, but instead because of invalid user names that contain non-ASCII data. When you access OracleAS Wireless through an HDML browser (such as a Japanese KDDI device), and attempt a login with an invalid account, the non-ASCII data in the user name is displayed as corrupted data or is changed to another string. To fix this, input the correct user account information.

16.5.25 Configuration Assistant Hanging

Problem

Why does the Wireless Configuration Assistant hang while I use it against my 64-bit database?

Solution

The account is being locked out because too many incorrect username and password requests have been received. Before recreating a 10*g* database on the same machine as the original database, (using the same Service Name and SQL*Net port as the original database), shut down any middle tiers that are trying to connect to the database.

If you must keep the middle tier running while creating the new database, then set `failed_login attempts to UNLIMITED` as a workaround.

16.5.26 DTMF Attribute <SimpleMenu>

Problem

Is `<SimpleMenu>` included in this release?

Solution

The online MXML tag glossary omits the `DTMF` attribute, `<SimpleMenu>`. This attribute controls whether DTMF keys are assigned automatically to the first nine `<SimpleMenuItem> element`s. The DTMF attribute takes a boolean value (`true`, `false`). The default value is `false`.

16.5.27 Errors When Starting OracleAS Wireless Using the Oracle Process Management and Notification Utility (opmnctl)

Problem

I get the following message when starting OracleAS Wireless or the middle tier:

```
ias-component/process-type/process-set:
wireless/performance_server/perfmonitor_1001
Error
--> Process (pid=27446)
failed to start a managed process after the maximum retry limit
Log:

/apps/oracle/product/10gmid/opmn/logs/wireless~performance_server~perfmonitor_
1001~1
```

In addition, the associated `wireless~performance_server~perfmonitor_1001~1` log file contains the following:

```
--------
04/12/29 12:12:20 Start process
--------
iAS Wireless System Log Directory is /apps/oracle/product/10gmid/wireless/logs
PanamaServer perfmonitor_1001 failed to initialize
PanamaServer is exiting ...
```

The related log entry for `$ORACLE_HOME/wireless/logs/log.xml` contains the following:

```
<MESSAGE>
<HEADER>
<TSTZ_ORIGINATING>2005-01-03T12:47:26.639-08:00</TSTZ_ORIGINATING>
<ORG_ID>ORACLE</ORG_ID>
<COMPONENT_ID>WIRELESS</COMPONENT_ID>
<MSG_TYPE TYPE="ERROR"></MSG_TYPE>
<MSG_LEVEL>1</MSG_LEVEL>
<HOST_ID>iaswtest1.us.oracle.com</HOST_ID>
<HOST_NWADDR>128.32.25.161</HOST_NWADDR>
<PROCESS_ID>null-Thread[main,5,main]</PROCESS_ID>
<USER_ID>oracle</USER_ID>
</HEADER>
<PAYLOAD>
<MSG_TEXT>[main]
sysmgmt.server.SMAgent.initForPanamaServer(SMAgent.java:637)
*Can not find the valid ServerCategory for instance
perfmonitor_1001 on host iaswtest1.us.oracle.com it is not
configured!*</MSG_TEXT>
</PAYLOAD>
</MESSAGE>
```

Solution

This problem occurs when the hostname in the `/etc/host` file has been manually changed to `iaswtest1.us.oracle.com`, which was not the original hostname used when the middle tier was installed or when OracleAS Wireless was configured. To solve this problem, restore the original `etc/host` file.

To change the hostname for the middle tier installation, follow the directions on changing the hostname, domain name, or IP address in the *Oracle9i Application Server Administrator's Guide*.

> **Note:** When you change the hostname, the *OracleAS Wireless SSO Partner URL* changes to the new hostname. You must also update SSO with the new *OracleAS Wireless SSO Partner URL*.

16.5.28 Processes Cannot Start Because of Incorrect Database Connection Configuration

Problem

The OracleAS Wireless standalone processes and OC4J_Wireless Instance did not start and the log entries in `$ORACLE_HOME/wireless/logs/log.xml` associated with SQLEXCEPTIONS and ORA-XXXXX are as follows:

```
java.sql.SQLException: ORA-12545: [... ...]
java.sql.SQLException: ORA-03113: [... ...]
java.sql.SQLException: ORA-00600: [... ...]
java.sql.SQLException: ORA-12514: [... ...]
java.sql.SQLException: ORA-12520: [... ...]
... ...
```

Solution

Incorrect database connection configuration for OracleAS Wireless can prevent the processes from starting. Verify that the Oracle Application Server infrastructure, including the database server, Oracle Internet Directory (OID) and OC4J_Security are properly configured and running. To do this:

- Verify that the database server has been properly configured with the correct initial parameters (such as `PROCESSES`) and that the database server machine is not overloaded with the current configuration.

  > **Tip:** Increase the number of PROCESSES of the Oracle Database for multiple Oracle Application Server middle-tier instances pointing to the same infrastructure database.

- Verify that OID is running. You can find out the current state of OID using the Oracle Directory Manager tool to connect to the OID Server and browse the OracleAS Wireless associated entities. For more information see *Oracle Internet Directory Administrator's Guide*.

16.5.29 Virtual Host Mapping to Hide the ptg/rm URL

Problem

The default Wireless device portal URL is http://myhost:myport/ptg/rm. I would like to change it to http://myhost:myport.

Solution

You can use Apache's mod_rewrite to achieve your goal. Add a rule to the ORACLE_HOME/Apache/Apache/conf/httpd.conf file. Add the following two lines to the end of the file.

RewriteEngine On

RewriteRule ^/$ http://myhost:myport/ptg/rm [R,NS,L]

16.5.30 Granting Change Password Privilege

Problem

I get an error when I try to change any user's password (from the webtool).

Solution

The user passwords for all OracleAS components (including the Wireless server) are managed by the Oracle Internet Directory (OID) server. OracleAS administrators can use the OID DAS tool to change user passwords. For security reasons the Wireless server admin tools, by default, are not granted permission to change user passwords. If for convenience you would prefer to be able to change user paswords from the Wireless admin tools, then you must grant permissions to the Wireless component. To do so, set the ORACLE_HOME and JAVA_HOME environment variables and run the following script from the OracleAS middle tier (If you have multiple middle tiers then you must run this script only once from any middle tier):

ORACLE_HOME/wireless/bin/assignUserSecurityAdminsPrivilege.sh(.bat)

The script takes two parameters: the OID super user DN and the super user password.

After you run this script, restart the OID server and the OC4J_Wireless process on the middle tier (If you have multiple middle tiers, then you must restart the OC4J_Wireless process on all middle tiers).

16.5.31 Password Policy Management

Problem

Everything worked fine but suddenly I cannot log in to the Wireless device portal or admin tools. I didn't change anything. What is wrong?

Solution

The most common reason for this problem is that your user account was locked by the current password policies. Password polices are sets of rules that govern how passwords are used. For example, by default the user passwords expire after 60 days. An account is locked out after 10 login failures. For more details about password policies and how to solve any related issues please see "Password Policies in Oracle Internet Directory" in Oracle Internet Directory Administrator's Guide.

16.5.32 Broken Images

Problem

I don't see any (or some) images on my device.

Solution

The problem probably comes from the authoring language used in your application. Try using MobileXML without image adaptation for your application.

In this case the device will download the images directly from your web application. Very often there is a firewall between the device and your web application which prevents the device from accessing the images. If that is the case then you have two options:

- Use image adaptation in your application. Publish all images, used by your application, outside the firewall and use absolute URLs for the images. Please note that you must deploy only the images outside the firewall. The rest of your application, i.e. JSPs, Servelts, etc., can be still inside the firewall.

- Use MobileXML with image adaptation or XHTML for your application. The most common reason for this problem is misconfiguration of the Wireless URL parameters and specificaly the URL for the image adaptation server. Please check the value for this parameter and pay attention whether your Wireless server uses site or instance URL parameters.

16.6 Diagnosing OracleAS Wireless Server Problems

Check the log files to diagnose problems, as they include information necessary for administrators and support personnel to help you solve problems.

16.6.1 Debugging the Oracle Streams Dispatcher

To diagnose a problem with the Oracle Streams dispatcher, ensure that you set the log level to *NOTIFY*; Oracle Streams-related errors display in the log. For more information on setting the log level, see Section 3.4.2.

16.7 Viewing UTF-8 Pages in Localized Languages with Netscape 4.7 or Earlier

Some languages may not display properly if you use Netscape 4.7 or an earlier version. In some cases, characters may display as boxes. To fix this problem, configure the Netscape preferences as follows:

1. Select *Preferences* from the drop-down menu. The *Preferences* dialog appears.
2. From the *Category* tree, select **Fonts** to display the *Fonts* dialog.
3. In the *Fonts* dialog, select **Unicode** from the *For the Encoding* drop-down list.
4. From the *Variable Width Font* and *Fixed Width Font* drop-down lists, select the font that supports the preferred language. For example, if you select Chinese as your preferred language, select *MS Song* to view the page in Chinese.

From the Netscape tool bar, select **Edit**.

16.8 Oracle Workflow and Oracle Application Server Wireless

Oracle Workflow can be used with OracleAS Wireless to send notifications to mobile devices. Ensure that the version of Oracle Workflow is 2.6.3 or later. The current version of Oracle Workflow is included in the Oracle Application Server 10*g* release.

16.9 Re-Registering the OracleAS Wireless Portal Services URL Reference in OracleAS Portal

Installing OracleAS Wireless 10*g* Release 2 (10.1.2.02) against Oracle Application Server Wireless 9.0.2 re-registers the OracleAS Wireless portal service URL and might prevent OracleAS Portal from responding to mobile requests. To avoid this problem, you must re-register the URL as follows:

1. In Portal Builder, click the **Administer** tab.
2. In the *Services* portlet, click **Global Settings**.
3. Click the **Mobile** tab.

4. Enter the OracleAS Wireless URL in the *OracleAS Wireless 10g Wireless Portal Service URL* field.
5. Click **OK**.

For more information about configuring mobile settings and updating the Oracle Application Server Wireless service URL in OracleAS Portal, refer to the *Oracle Application Server Portal Configuration Guide.*

16.10 Information on HTTPS Support

For information on supporting HTTPS between a mobile device and the OracleAS Wireless and Voice Portal (`ptg/rm`) as well as between the OracleAS Wireless and Voice Portal and applications, see the following:

- Chapter 10, "OracleAS Wireless Security"
- Chapter 11, "Mobile Single Sign-On"
- *Oracle Application Server Security Guide*
- *Oracle Application Server Single Sign-On Administrator's Guide*

16.11 Configuring OracleAS Wireless for Load-Balancing

After the load-balancer is installed and configured, you must configure the first OracleAS Wireless middle tier to use the site URLs as described in Section 3.9.1.1.2. The site URLS must point to the load balancer. Configure the remaining OracleAS Wireless middle tiers to use the instance URLs as described in Section 3.5.

16.12 Need More Help?

You can find more solutions on Oracle *MetaLink* http://metalink.oracle.com. If you do not find a solution for the problem, log a service request.

> **See Also:** *Oracle Application Server Release Notes*, available on the Oracle Technology Network:
> http://www.oracle.com/technology/documentation/index.html

Glossary

access points

An access point is the address monitored by the **Async Listener**. Using the System Manager, you can create the following type of access points.

- **Site Access Point** —Enables access to all the asynchronous applications.
- **Application Category Access Point** — An access point associated with one or more application link categories. See also **application link category**.

actionable message

An interactive push message sent from the Oracle Application Server Wireless instance to a user's device. Actionable messages are sent through SMS, e-mail and Instant Messaging and can be acted upon by users. Other, non-actionable messages are in final form once delivered to a user's device, prohibiting users from replying to these messages.

adapter

A dynamically loaded Java class that acquires content from an external source, such as a Web site or a database, and converts the content into Mobile XML. Pre-built adapters include the Web Integration adapter, SQL adapter, and Strip adapter.

Adapter Result format

A general, user interface-independent content format. Content in Adapter Result format requires conversion to Simple Result format before it can be converted to the final target format.

antenna

Each **tag** has at least one antenna. On the other side of the communication link, the **reader** must also have an antenna. Some readers can drive multiple antennae at the same time. Depending on the protocol, frequency and application, these antennae vary from thin strips of metal laid across a surface, to a portal doorway antenna that is meters tall

application link

Sometimes referred to as an application, an application link is pointer to a master application. The master application defines the core properties of the application link. Using application links, Content Managers customize and publish master applications.

application link category

Application Link Categories, which are sets of applications, support **Premium SMS** and **Reverse Charge SMS**. For example, in PremiumSMS, each set of applications having the same premium level can be put into an application link category.

Async Listener

The Async Listener, a client of the **Messaging Server**, interprets and processes the e-mail and SMS requests. See also **Premium SMS** and **Reverse Charge SMS**.

bookmark

A link from a service to an external, device-compatible data source that does not require OracleAS Wireless processing.

chip

A silicon chip, with embedded memory, is used in a **tag**. The chip implements the wireless protocol and access functions to its embedded memory. Note that in active tags, this is not a single chip but an entire board.

Customization Portal

A Web-based interface (also referred to as the Wireless Customization Portal) that end users access to select services and configure their device portal. Users access the Customization Portal from their desktop computers.

daemon

A background process that performs a specified operation in response to certain events or at specified times.

device

An object that describes either a physical device, such as a cellular phone, or an application, such as email. There is a default device transformer for each device. For the Oracle Sensor Edge Server, a device is an end point of a sensor-based architecture, such as an RFID **reader**, a dry contact, a laser diode, carousel, or a robotic picker.

device transformer

A transformer that converts content from Simple Result format into the target format.

DOM Interface

Document Object Model. The interface that allows programs and scripts to access and transform processed XML documents.

DTD

Document Type Definition. A file in an XML document that defines how the application presenting the document should interpret the XML document.

end user

A person who accesses a OracleAS Wireless service from a client device.

HDML

Handheld Device Markup Language. A reduced version of HTML designed to enable wireless pagers, cellular phones, and other handheld devices to access Web page content.

IMAP

Interactive Mail Access Protocol. A hierarchical mail storage and retrieval structure.

HTML

HyperText Markup Language. The document format that defines the page layout, fonts, and graphic elements, as well as the hypertext links to other documents on the Web.

JNDI

Java Naming and Directory Interface. A set of APIs that provide directory and naming functionality to Java applications.

JSP

JavaServer Pages. A technology based on Java servlets which separates the functions of Web page layout and content generation. JavaServer Pages technology enables the creation of server-generated Web pages incorporating dynamic content.

LDAP

Lightweight Directory Access Protocol. Protocols for accessing directories. The LDAP protocols support TCP/IP.

master application

The core implementation of an OracleAS Wireless application. The master application invokes a specific adapter, and identifies the transformer used to convert content for the target device.

Messaging Server

The Messaging Server is the component that delivers messages and notifications by interfacing with the **Async Listener** and **Notification Engine** and a messaging gateway to deliver messages.

MIME

Multipurpose Internet Mail Extensions. A mail type that defines the message structure for different 8-bit character sets and multi-part messages.

Mobile Portal

The interface where mobile device users access their Oracle Application Server Wireless applications.

Notification Engine

The Notification Engine, a client of the **Messaging Server**, processes the notifications dequeued from the Notification Event Collector and sends them to the Messaging Server for ultimate delivery to users.

Oracle Application Server Wireless XML

A set of DTDs and XML document conventions used by the OracleAS Wireless to define content and internal objects.

Oracle Sensor Edge Server

The server that resides between all of the readers and the application middle tier. It is responsible for interfacing with all of the readers and sending normalized data back to the application server.

Premium SMS

The **Async Listener** enables users of SMS-enabled phones to access content from the Internet. To request such an application, a mobile user sends a message containing SMS keywords describing the application to an Async account using a short address (a number) known as the Large Account. The SMS keywords identify the application (for example, *ST* for stock quote applications.) The message goes through the network of a PremiumSMS operator to retrieve the content supplied by the Content Provider, whose system listens for the SMS message sent to the Large Account. The Content Provider processes the message and returns the requested information as a message to the user, who is charged a premium on top of the standard SMS transport rate for mobile device-issued requests. The content provider and Premium SMS operator (or carrier) both share this premium.

provisioning adapter

The adapter used to create, modify, and delete user objects in the OracleAS Wireless repository.

Radio Frequency Identification (RFID)

RFID is the use of small transponders with embedded Electronic Serial Numbers (ESNs) or memory, which transmit identifiers across one or more frequencies.

read

The process of retrieving data stored on an **Radio Frequency Identification (RFID)** tag by sending radio waves to the **tag** and then converting the waves the tag sends back into data is known as a read.

reader

A reader reads from, and writes to, the **tag** (or tags) to which it is connected. Readers usually have serial interfaces used to communicate with a host computer. There is no widely-accepted standard for this protocol.

reader field

The area of coverage for a reader. If tags are outside of a reader field, then they cananot receive radio waves and cannot be read. See also **tag**.

Real Time Location System (RTLS)

A technology that uses radio-frequency to produce real-time location information for tagged items.

repository

An Oracle database which stores all of the OracleAS Wireless objects, such as users, groups, adapters, and applications.

request

A query to initiate a desired OracleAS Wireless service. Requests are submitted on behalf of end-users to the OracleAS Wireless server.

request manager

The OracleAS Wireless component that processes requests for services. The request manager authenticates the user, submits the request to the OracleAS Wireless core, and retrieves the device type and any presentation settings. The request manager also forwards converted content from the transformer to the user.

request object

An XML document representing a request for service.

result transformer

A transformer that converts content from **Adapter Result format** format into **Simple Result format**.

Reverse Charge SMS

Reverse Charge is a billing model which charges the service premium to the mobile subscriber on the result SMS message, rather than on the service request itself. Mobile users, requesting applications through multiple channels, such as IVR (interactive voice response) or the Web, receive the service result as an SMS message. For example, when a user wants to access an article on the Web, the user must first complete and submit a web form requesting his SMS address before receiving an SMS message containing the authorization code needed to access the article. In this case, the user is charged a transport fee and a service premium for the SMS result message conveying the authorization code.

Usually with SMS, the sender of an SMS message is charged. With ReverseCharge, however, the party receiving the message is charged a transport fee and a service premium. The amount of the service premium depends upon which service the mobile user requests; each service has its own associated tariff class. To ensure the correct billing information, the application provider supplies the ReverseCharge operator with the Large Account and the tariff class of the service upon generating the service result SMS message.

RMI

Remote Method Invocation. A standard for creating and calling remote objects. RMI allows Java components stored in a network to be run remotely.

sample repository

The initial OracleAS Wireless repository, which includes pre-built objects such as transformers, adapters, and devices.

Service Manager

The visual interface for creating and managing OracleAS Wireless users, user groups, adapters, transformers, and services.

short name

A site-wide, unique name that identifies an OracleAS Wireless application. Device users invoke applications by sending messages to the site address with short names for the requested applications in the body or subject line of the message. For example, a user requests a stock quote application by sending a message to a site address (such as *ask@oraclemobile.com*) with the short name of the stock quote application (*stk*) in the body of the message.

Simple Result format

A content format that contains abstract user interface elements such as text items, menus, forms, and tables.

source format

The original format of content retrieved from an external data source by a OracleAS Wireless adapter. For example, the source format of Web page content is HTML.

SQL adapter

An adapter that retrieves and adapts content from any JDBC-enabled data source.

Strip adapter

An adapter that retrieves and adapts Web content dynamically.

strip level

The class used by the strip adapter to process markup tags in source content.

stylesheet

An XSLT (eXtensible Stylesheet Language Transformations) instance that implements content presentation for XML documents. OracleAS Wireless transformers can be either XSLT stylesheets or Java programs.

tag

(Also known as an RFID tag.) A single unit that contains a chip, one or more antennae, and a power source. If it is battery-driven or from a external source, the tag is an Active Tag. If the power source is inductive-based (which means that it relies on photoelectric effect to generate power from remotely generated radio waves), the tag is a Passive Tag. A tag containing data that cannot be changed is a read-only tag.

TTML

Tagged Text Mark-up Language. A lightweight version of HTML suitable for most PDAs.

target format

The format required to deliver data to a specific type of client device.

Thin HTML

A minimal version of HTML implemented by a transformer in the starter OracleAS Wireless repository. Thin HTML does not include support for frames, JavaScript, or other advanced features.

transformer

A OracleAS Wireless object that converts content returned by the OracleAS Wireless adapters. Result transformers convert Adapter Result documents into Simple Result documents. Device transformers convert Simple Result documents into the target format.

user agent

An object that associates an end user with a device type.

user group

A OracleAS Wireless object that represents a set of users that are grouped together based on common criteria such as interests, subscription level, or geographic location.

VoxML

A markup language that enables the use of voice to interface with applications.

WAP

Wireless Application Protocol. A wireless standard from Motorola, Ericsson, and Nokia for providing cellular phones with access to email and text-based Web pages. WAP uses Wireless Markup Language (WML).

Web Integration adapter

An adapter that retrieves and adapts Web content using WIDL files to map the source content to OracleAS Wireless XML.

WIDL

Web Interface Definition Language. A meta-data language that defines interfaces to Web-based data and services. WIDL enables automatic and structured Web access by compatible applications.

WIDL file

A file written in Web Interface Definition Language that associates input and output parameters with the source content that you want to make available in a OracleAS Wireless service.

WML

Wireless Markup Language. A markup language optimized for the delivery of content to wireless devices.

XML

eXtensible Markup Language. A flexible markup language that allows tags to be defined by the content developer. Tags for virtually any data item can be created and used in specific applications, allowing Web pages to function like database records.

XSLT

Extensible Stylesheet Language Transformations. A language for transforming one XML DTD into another XML DTD.

Index