

Oracle9i Application Server Wireless Edition

Implementation Guide

Release 1.1

January 2001

Part No. A86699-01

ORACLE®

Part No. A86699-01

Copyright © 2001 Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

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

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

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

Oracle is a registered trademark, and JDeveloper, Oracle8, Oracle8i, PL/SQL, SQL*, and SQL*Plus are trademarks or registered trademarks of Oracle Corporation. All other company or product names mentioned are used for identification purposes only and may be trademarks of their respective owners.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc., in the U.S. and other countries.

Contents

Send Us Your Comments	xvii
Preface	xix
1 Introduction	
1.1 New Features	1-1
1.2 What is the Wireless Edition?.....	1-2
1.2.1 Form Factor Considerations	1-3
1.2.2 User's View	1-3
1.3 How Wireless Edition Works	1-4
1.3.1 Alerts.....	1-5
1.4 Location-Based Services	1-6
1.5 Wireless Edition Architecture.....	1-6
1.5.1 Content Acquisition.....	1-6
1.5.1.1 Adapters.....	1-6
1.5.1.2 Services and Master Services.....	1-9
1.5.1.3 Service Manager.....	1-10
1.5.2 Content Delivery	1-10
1.5.3 Core Processing.....	1-11
1.5.3.1 Service Designer.....	1-12
1.5.3.2 Personalization Portal	1-13
1.6 Extensibility: Adding Adapters and Transformers.....	1-14

2 Wireless Edition Tools

2.1	Overview.....	2-1
2.2	Development Tools	2-1
2.2.1	Service Designer	2-1
2.2.1.1	Master Service Creation Wizard	2-4
2.2.2	The Transformer Testing Tool.....	2-4
2.2.3	Region Modeling Tool	2-6
2.2.4	The Web Integration Developer	2-7
2.2.5	XML Editor	2-7
2.2.6	Oracle XML Parser	2-8
2.3	Personalization.....	2-8
2.3.1	Personalization from Desktop	2-8
2.3.2	Personalization from Device.....	2-9
2.4	Oracle Performance Manager	2-9
2.4.1	Accessing the Oracle Performance Manager.....	2-10
2.4.2	Logging into the Oracle Performance Manager.....	2-10
2.4.3	Performance Data Collection Schema	2-12
2.4.3.1	PTG_SERVICE_LOG	2-12
2.4.3.2	PTG_SESSION_LOG.....	2-13
2.4.3.3	SERVICE_REQUEST_STATS	2-14
2.4.3.4	SERVICE_ERROR_STATS	2-14
2.4.4	Performance Statistics.....	2-15
2.4.4.1	Session Duration Statistics.....	2-15
2.4.4.2	Session Error Statistics.....	2-15
2.4.4.3	Session Service Statistics	2-15
2.4.4.4	Service Error Statistics	2-15
2.4.4.5	User Service Statistics	2-16
2.4.4.6	Service Invocation Statistics.....	2-16
2.4.4.7	Service Response Time Statistics.....	2-16
2.4.5	Fault Management	2-16
2.4.6	Configuration Management.....	2-16

3 Wireless Edition Services

3.1	Overview.....	3-1
3.2	Using the Service Designer	3-3

3.2.1	Starting the Service Designer	3-4
3.2.1.1	RMI Connection	3-5
3.2.2	Creating Objects	3-6
3.2.3	Modifying Objects.....	3-6
3.2.4	Deleting Objects.....	3-7
3.2.5	Flagged Objects	3-7
3.2.6	Refreshing the Repository View	3-7
3.2.7	Object Identifiers	3-8
3.3	Creating a Master Service.....	3-8
3.3.1	Step 1: Selecting or Creating a Service Folder.....	3-9
3.3.2	Step 2: Setting the Master Service Properties.....	3-12
3.3.3	Step 3: Selecting an Adapter	3-14
3.3.4	Step 4: Setting the Init Parameters.....	3-14
3.3.5	Step 5: Setting the Input Parameters	3-14
3.3.6	Step 6: Setting the Output Parameters.....	3-16
3.3.7	Step 7: Creating the Result Transformer (Optional)	3-18
3.3.8	Step 8: Setting the Device Transformer (Optional)	3-18
3.4	Creating a Location-Based Service.....	3-19
3.4.1	Using the Service Designer to Create Location-Based Services	3-19
3.4.2	Using the Master Service Creation Wizard to Create Location-Based Services .	3-20
3.5	Modifying a Master Service	3-21
3.5.1	General Panel.....	3-21
3.5.2	Init Parameters Panel.....	3-21
3.5.2.1	Web Integration Init Parameters.....	3-22
3.5.2.2	SQL Adapter Init Parameters.....	3-22
3.5.2.3	URL Adapter Init Parameters	3-23
3.5.2.4	Servlet Adapter Init Parameters	3-24
3.5.2.5	Mail Adapter Init Parameters.....	3-25
3.5.2.6	Directory Adapter Init Parameters.....	3-25
3.5.3	Input Parameters Panel.....	3-26
3.5.3.1	Web Integration Input Parameters.....	3-28
3.5.3.2	Mail Adapter Input Parameters.....	3-30
3.5.3.3	SQL Master Service Input Parameters.....	3-31
3.5.3.4	Stripper Adapter Input Parameters.....	3-31
3.5.3.5	URL Adapter Input Parameters.....	3-32

3.5.3.6	Servlet Adapter Input Parameters	3-33
3.5.4	Output Parameters Panel	3-34
3.5.5	Result Transformer Panel.....	3-34
3.5.6	Device Transformer Panel.....	3-34
3.6	Deleting a Master Service	3-35
3.7	Creating a Folder	3-35
3.8	Deleting a Folder	3-37
3.8.1	Service Trees.....	3-37
3.9	Creating a Service Alias.....	3-38

4 Using the Region Modeling Tool

4.1	Overview.....	4-1
4.1.1	The System-Defined Region Hierarchy	4-2
4.1.2	User-Defined Regions	4-6
4.2	Accessing the Region Modeling Tool	4-6
4.3	Managing Spatial Objects.....	4-6
4.3.1	Accessing the Region Modeling Tool in the Standalone Mode.....	4-7
4.3.2	Creating User-Defined Regions.....	4-7
4.3.2.1	Creating a User-Defined Region from the SDR Hierarchy	4-7
4.3.2.2	Creating a UDR from an Address.....	4-9
4.3.3	Deleting a User-Defined Region	4-10
4.3.4	Creating a Folder	4-11
4.3.5	Deleting a Folder	4-12
4.3.6	Creating System-Defined Regions	4-12
4.3.6.1	Loading SDR Data Using SQL*Plus	4-12
4.3.6.2	Loading SDRs from the Region Modeling Tool	4-13
4.3.6.2.1	Adding Data Under a Continent	4-14
4.3.6.2.2	Adding Data Under a Country	4-15
4.3.6.2.3	Adding State Data Under a Country	4-15
4.3.6.2.4	Adding Data for the Postal Code, County, or City.....	4-16
4.3.7	Deleting a System-Defined Region.....	4-17
4.4	Creating Location-Based Master Services and Folders.....	4-17
4.4.1	Accessing the Region Modeling Tool in the Browse Mode	4-17
4.4.1.1	Modifying a Folder or Service to be Location-Based.....	4-18
4.4.1.2	Using the Service Designer to Create a Location-Based Folder or Service...	4-18

4.4.1.3	Using the Master Service Creation Wizard to Create a Location-Based Service or Folder	4-19
4.5	Creating Location-Based Services with Location-Specific Content	4-20
4.5.1	Accessing the Current Subscriber's Location Using the Runtime API	4-20
4.5.2	Accessing the Current Subscriber's Location Using Derived Location-Related System Parameters	4-20

5 Managing Users and Groups

5.1	Overview	5-1
5.2	Managing Groups Using the Service Designer	5-1
5.2.1	Creating User Groups Using the Service Designer	5-2
5.2.2	Deleting Groups Using the Service Designer	5-4
5.2.3	Adding Users to a Group Using the Service Designer	5-4
5.2.4	Adding Services to a Group Using the Service Designer	5-4
5.2.5	Removing Users from a Group Using the Service Designer	5-5
5.2.6	Removing Services from a Group Using the Service Designer	5-6
5.3	Managing Groups and Users Using the Personalization Portal	5-6
5.3.1	Accessing the Personalization Portal	5-7
5.3.2	Logging in as an Administrator	5-8
5.3.3	Using the Administration Tab	5-9
5.3.3.1	Finding Users	5-10
5.3.3.2	Creating Users	5-11
5.3.3.3	Editing User Properties	5-14
5.3.3.4	Assigning a User to a Group	5-17
5.3.4	Managing Groups Using the Personalization Portal	5-19
5.3.4.1	Creating a Group Using the Personalization Portal	5-19
5.3.4.2	Managing Groups	5-19
5.3.4.3	Deleting a Group	5-20
5.3.4.4	Adding Users to a Group	5-21
5.3.4.5	Assigning Services to a Group	5-22
5.4	Integrating Users with Existing Provisioning Systems	5-23
5.4.1	ProvisioningHook Interface	5-23
5.4.2	UserAuthenticationHook Interface	5-24

6 Personalization from Desktops

6.1	Overview.....	6-2
6.2	Logging in to Personalize the Portal.....	6-2
6.3	Logging In as a User.....	6-5
6.4	Activating a Service for a User	6-5
6.4.1	Copying or Editing a Service	6-6
6.4.1.1	Configuring Input Arguments and Conditions.....	6-8
6.5	Editing a Folder	6-8
6.5.1	Reordering Folder Contents	6-8
6.6	Creating a Subfolder	6-10
6.6.1	Moving Objects Between Folders.....	6-10
6.7	Creating a Bookmark	6-10
6.8	Creating Alerts.....	6-11
6.8.1	Setting Alert Parameters.....	6-12
6.8.1.1	The Alert Sent Every Settings.....	6-12
6.8.1.2	The Alert Start Time settings.....	6-12
6.8.2	Editing Alert Parameters.....	6-13
6.9	Managing Alert Addresses	6-14
6.9.1	Creating a New Alert Address.....	6-15
6.10	Creating a LocationMark.....	6-16
6.11	Editing a LocationMark	6-19
6.12	Changing the Default Status of a LocationMark.....	6-19
6.13	Deleting a LocationMark.....	6-19
6.14	Editing a User Profile.....	6-20

7 Personalization from a Device

7.1	Overview.....	7-1
7.2	Device Personalization.....	7-2
7.2.1	Navigation and Data Entry	7-2
7.3	Logging in to the Portal	7-3
7.4	Creating a Folder	7-3
7.5	Creating a Bookmark	7-4
7.6	Reordering Objects	7-4
7.7	Renaming a Folder	7-5
7.8	Showing or Hiding Objects.....	7-5

7.9	Deleting a Folder	7-6
7.10	Editing a User Profile.....	7-7
7.10.1	Change Password.....	7-7
7.10.2	Assigning an Alert Address	7-7
7.10.3	Activating Autolocate.....	7-8
7.11	Creating a LocationMark.....	7-8
7.12	Logging Off	7-9

8 Logical Devices

8.1	Overview	8-1
8.2	Creating a Logical Device	8-2
8.3	Modifying a Logical Device.....	8-5
8.4	Deleting a Logical Device.....	8-6
8.5	Determining the Client Device Type.....	8-6

9 Walkthroughs

9.1	Purpose	9-1
9.2	Before You Begin	9-2
9.2.1	Creating a User Group and a User	9-3
9.3	Creating a SQL Service	9-5
9.3.1	Task 1: Using the Master Service Creation Wizard.....	9-5
9.3.2	Task 3: Creating an Alias that Points to SQL Time	9-6
9.3.3	Task 4: Making SQL Time Available to a Group.....	9-7
9.3.4	Task 5: Testing SQL Time	9-8
9.4	Creating a PL/SQL Service.....	9-9
9.4.1	Task 1: Loading the PL/SQL Package, demo_proc.sql	9-9
9.4.1.1	Sample Code Explanation.....	9-12
9.4.2	Task 2: Creating the Master Service	9-13
9.4.3	Task 3: Creating an Alias that Points to PLSQL Test.....	9-15
9.4.4	Task 4: Making PLSQL Test Available to a Group.....	9-16
9.4.5	Task 5: Testing PLSQL Test	9-16
9.5	Creating a Master Service Using the URL Adapter	9-18
9.5.1	Task 1: Building the Java Program	9-18
9.5.1.1	Java Example Overview	9-23
9.5.1.2	Running the Java Example	9-24

9.5.2	Task 2: Creating the Master Service.....	9-28
9.5.3	Task 3: Creating an Alias that Points to URL Adapter	9-29
9.5.4	Task 4: Making URL Example Available to a Group.....	9-29
9.5.5	Task 5: Testing URL Example	9-30
9.6	Creating Location-Based Services	9-31
9.6.1	Task 1: Creating the Location-Dependent Master Service	9-31
9.6.2	Task 2: Creating an Alias that Points to Geocoder	9-33
9.6.3	Task 3: Making Geocoder Available to a Group.....	9-33
9.6.4	Task 4: Testing the Geocoder	9-34
9.7	Creating the Spatial Adapters.....	9-35
9.8	Creating a Master Service Using the Web Integration Adapter	9-36
9.8.1	Task 1: Creating the WIDL File in the Web Integration Developer.....	9-37
9.8.2	Task 2: Publishing the WIDL Interface to the Web Integration Server	9-39
9.8.3	Task 3: Using the Service Designer.....	9-40
9.8.4	Task 4: Creating the Master Service.....	9-41
9.8.5	Task 5: Creating an Alias that Points to Stock Quotes.....	9-42
9.8.6	Task 6: Making Stock Quotes Available to a Group	9-43
9.8.7	Task 7: Testing Stock Quotes	9-43
9.9	Creating Chained Services Using the Web Integration Developer	9-45
9.9.1	Task 1: Creating the First Service in the Chain	9-46
9.9.2	Task 2: Creating the Second Service in the Chain	9-47
9.9.3	Adding the Service to the Wireless Edition Repository	9-49
9.9.4	Creating a User and an Alias and Test the Service.....	9-51
9.10	Creating External Links Using the Service Designer	9-52

A The Mail Adapter

A.1	Overview.....	A-1
A.1.1	Message Management Functions.....	A-1
A.1.2	Folders Management Functions.....	A-2
A.2	The End User Interface	A-2
A.3	Configuration Parameters for the Mail Adapter.....	A-10

Glossary

Index

List of Figures

1-1	The Role of the Wireless Edition	1-3
1-2	The Wireless Edition Device Portal	1-3
1-3	Isolating Content Acquisition from Content Delivery	1-4
1-4	Processing a User Request.....	1-5
1-5	The Wireless Edition Server.....	1-12
1-6	The Service Designer.....	1-13
1-7	The Personalization Portal	1-14
2-1	The Service Designer.....	2-2
2-2	The Welcome Screen of the Master Service Creation Wizard.....	2-4
2-3	The Transformer Testing Tool	2-5
2-4	The Region Modeling Tool.....	2-6
2-5	The XML Editor	2-8
2-6	The Personalization Portal	2-9
2-7	The Oracle Performance Manager	2-10
2-8	The Login Screen of the Oracle Performance Manager	2-11
2-9	The Navigator Tree	2-12
2-10	Using the Oracle Enterprise Manager Console to Increment Pool Size	2-17
3-1	The Service Designer.....	3-4
3-2	Flagged Objects in the Repository Tree.....	3-7
3-3	The Select Folder Screen	3-9
3-4	The Create New Folder Screen.....	3-10
3-5	The Setting Service Properties Screen.....	3-12
3-6	The Selecting Adapter Screen	3-14
3-7	The Init Parameters Panel for the Servlet Adapter.....	3-24
3-8	The Input Parameters Panel for the Web integration Adapter.....	3-29
3-9	The Input Parameters Panel for the Stripper Adapter	3-31
3-10	The Input Parameters Panel for the Servlet Adapter	3-33
3-11	The Create New Folder Form	3-36
3-12	Create New Alias Form (First Form)	3-39
4-1	The Region Modeling Tool.....	4-2
4-2	Hierarchy of the System-Defined Region	4-3
4-3	The Region Modeling Tool with Region Menu Displayed.....	4-7
4-4	The Create UDR Form	4-8
4-5	The User-Defined Region	4-9
4-6	The Create From Address Screen.....	4-10
4-7	The Create Folder Screen.....	4-11
4-8	The Add Country Data Form.....	4-14
4-9	The Add State Data Form.....	4-16
4-10	The Creating Folder Screen of the Master Service Creation Wizard	4-19
5-1	Create New Group Form.....	5-2

5-2	The Service Folders Tab.....	5-5
5-3	The Services screen of the Personalization Tab	5-8
5-4	The User Page of the Personalization Portal	5-9
5-5	The New User Page.....	5-12
5-6	New User Confirmation.....	5-14
5-7	The Found User List.....	5-15
5-8	The User Profile Page.....	5-15
5-9	The Assign Group to User page.....	5-18
5-10	The Groups Page	5-20
5-11	The Found User List.....	5-21
5-12	The Assign Group to User Page.....	5-21
5-13	The Select Service for Group Page.....	5-22
6-1	The Login Screen of the Personalization Portal	6-3
6-2	The Service Subscription Page.....	6-4
6-3	The Personalization Portal Icons.....	6-6
6-4	The Copy Service Page	6-7
6-5	The Reorder Services List Box	6-9
6-6	The Create Bookmark Screen.....	6-11
6-7	The Create Alert Screen.....	6-13
6-8	The Alert Address Page.....	6-15
6-9	The Create LocationMark Page.....	6-17
6-10	The LocationMark List Page.....	6-19
6-11	The User Profile Page.....	6-20
7-1	Home Screen for Device User	7-2
7-2	Adding a Folder.....	7-4
7-3	Reordering Objects	7-5
7-4	Hidden Objects	7-6
7-5	Hide / Show State	7-6
7-6	The User Profile Screen.....	7-7
7-7	Creating an Alert Address	7-8
7-8	Adding a LocationMark	7-9
8-1	The Logical Device Name and Display Data Form.....	8-2
8-2	The Login Page Form.....	8-4
8-3	The Error Page Form.....	8-5
9-1	The Init Parameters screen for the SQL Adapter.....	9-15
9-2	The Create New Bookmark Form	9-52
A-1	The Login Page	A-3
A-2	The Inbox Page	A-3
A-3	The Actions Page.....	A-4
A-4	The Compose Message Page.....	A-5
A-5	The Folder Listing Page.....	A-5

A-6	The Folder Page	A-6
A-7	The Create Folder Page	A-7
A-8	The Read Message Page.....	A-8
A-9	The Reply Message Page	A-8
A-10	The Forward Message Page	A-9
A-11	The Move Message Page	A-10

List of Tables

2-1	Objects in the Wireless Edition Repository	2-2
2-2	PTG_SERVICE_LOG.....	2-12
2-3	PTG_SESSION_LOG.....	2-13
2-4	SERVICE_REQUEST_STATS.....	2-14
2-5	SERVICE_ERROR_STATS	2-14
3-1	Parameters for the Create New Service Folder Screen	3-10
3-2	Configuration Properties for the Master Service	3-12
3-3	Attributes of the Wireless Edition Adapters	3-15
3-4	Output Parameters for the Wireless Edition Adapters.....	3-16
3-5	Init Parameters of the Web Integration Adapter	3-22
3-6	Init Parameters for the SQL Adapter.....	3-23
3-7	Init Parameters for the Servlet Adapter	3-25
3-8	Init Parameters for the Directory Adapter.....	3-25
3-9	Input Parameters	3-26
3-10	Input Parameters Attributes	3-27
3-11	Init Parameters for the Mail Adapter	3-30
3-12	Input Parameters for the Stripper Adapter	3-31
3-13	Input Parameters for the URL Adapter.....	3-33
3-14	Input Parameters for the Servlet Adapter	3-34
3-15	Parameters for the Create New Folder Form	3-36
3-16	Parameters for the Create New Alias Form.....	3-39
4-1	The Region Model Data Schema	4-4
4-2	Example of mystate Table	4-12
4-3	Fields in the Add Data Screens	4-14
4-4	The Derived Location-Related System Parameters.....	4-20
5-1	Parameters of the Create New Group Form.....	5-2
5-2	Login Elements of the Service Tab of the Personalization Portal.....	5-8
5-3	Elements of the Found User List	5-11
5-4	Parameters for the New User Page.....	5-12
5-5	Parameters for the User Profile Page.....	5-16
5-6	Elements of the Assign Group to User Screen	5-18
5-7	Elements in the Group List Section of the Groups Page.....	5-20
6-1	Login Elements of the Service Tab of the Personalization Portal.....	6-3
6-2	The Personalization Portal Tabs.....	6-4
6-3	Personalization Portal Icon Functions.....	6-6
6-4	Elements in the LocationMark List.....	6-18
6-5	Parameters for the User Profile Page.....	6-20
8-1	Parameters for the Create New Device Screen	8-2
8-2	Parameter of the Login Page.....	8-4

8-3	Parameter of the Error Page.....	8-5
9-1	Configuration Parameters	9-24
9-2	Servlet Parameters.....	9-27
9-3	Parameters for the New Service Dialog	9-38
9-4	Parameters of the New Service Dialog.....	9-46
9-5	Parameters for the New Service Dialog	9-48
A-1	Options in the Actions Page.....	A-4

Send Us Your Comments

Oracle9i Application Server Wireless Edition Implementation Guide, Release 1.1

Part No. A86699-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: iasdocs_us@oracle.com
- Postal service:
Oracle Corporation
Oracle Mobile and Wireless Products
500 Oracle Parkway, Mailstop 4OP6
Redwood Shores, California 94065
USA

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

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

Preface

The Oracle9i Application Server Wireless Edition Implementation Guide discusses how you can use the Wireless Edition to develop and deliver mobile services. The guide includes these chapters:

Chapter 1, "Introduction"	Provides an overview of the Wireless Edition.
Chapter 2, "Wireless Edition Tools"	Introduces the system development and management tools you can use to implement the Wireless Edition.
Chapter 3, "Wireless Edition Services"	Describes how to create and manage the Wireless Edition services.
Chapter 4, "Using the Region Modeling Tool"	Describes how to use the Region Modeling Tool to create spatial objects and assign them to master services.
Chapter 5, "Managing Users and Groups"	Describes how to use the Service Designer and the Personalization Portal to create users and groups.
Chapter 6, "Personalization from Desktops"	Describes how users can personalize their service using the Personalization Portal from a desktop.
Chapter 7, "Personalization from a Device"	Describes how users can personalize their service using the Personalization Portal from a device.

Chapter 8, "Logical Devices"

Describes how to create and manage the Wireless Edition objects that represent the target platforms and devices.

Chapter 9, "Walkthroughs"

Provides step-by-step walkthroughs for implementing basic Wireless Edition services.

Introduction

This document introduces the Oracle9i Application Server Wireless Edition. Each section of this document presents a different topic. These sections include:

- [Section 1.1, "New Features"](#)
- [Section 1.2, "What is the Wireless Edition?"](#)
- [Section 1.3, "How Wireless Edition Works"](#)
- [Section 1.4, "Location-Based Services"](#)
- [Section 1.5, "Wireless Edition Architecture"](#)
- [Section 1.6, "Extensibility: Adding Adapters and Transformers"](#)

1.1 New Features

The Wireless Edition includes the following new features:

Service Designer Enhancements

- Service creation wizard.
- Region modeling tool.
- Transformer creation and test tool.
- Location dependent services.

Wireless Edition Core

- Multiple JVM support.
- Enhanced Wireless Edition API including runtime and data model.
- Enhanced URL Adapter that supports parameters and cookies.

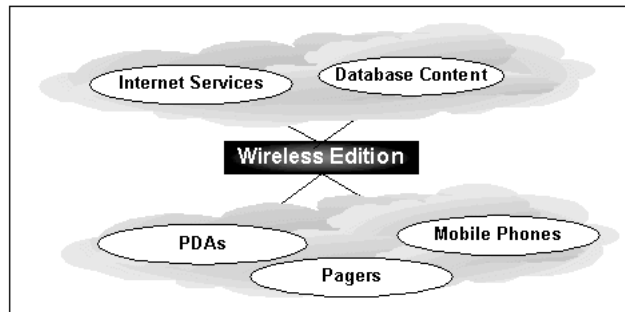
- Enhanced Directory (LDAP, X.500, JNDI) and Email (IMAP4 and POP3) Adapters.
- Improved handling of multibyte character sets.
- Enhanced voice gateway transformers (VoxML and VoiceXML).
- Support location-based services including routing, yellow pages, geocoding, location acquisition.
- Integration with Oracle Enterprise Manager.
- Enhanced logging and event management.
- Enhanced Wireless Edition DTD.

Personalization Portal

- Support personalization from devices (PAPZ Lite).
- Enhanced Wireless Edition personalization API.
- Improved wireless Personalization Portal that enables subscribers to create locationmarks and alerts.
- Subscriber management (i.e., create, modify, and delete subscribers from Web-based Personalization Portal).

1.2 What is the Wireless Edition?

The Wireless Edition is a portal service for delivering information and applications to mobile devices. The Wireless Edition makes Web and database applications, such as email, news, and directory services, accessible to mobile device users without having to rewrite content for every target platform.

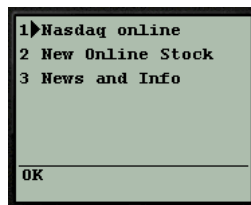
Figure 1–1 The Role of the Wireless Edition

1.2.1 Form Factor Considerations

Most Internet content is not intended to be viewed or accessed from mobile devices, which typically have limited display and inputting capabilities. Many attempts to provide Web content to mobile devices either do not consider the form factor of a mobile device, or require developers to create new content and applications for each platform that they want to support. The Wireless Edition avoids these shortcomings by dynamically converting content into the format and form factor of the target device. Using the Wireless Edition, developers write Web content once and distribute it to a variety of wireless devices, such as pagers, personal digital assistants, and mobile phones.

1.2.2 User's View

End users of the Wireless Edition access mobile services from the Wireless Edition device portal. The device portal displays a list of Wireless Edition-generated services and folders. Users can configure their device portals by selecting the services that they want to access by creating bookmarks and by organizing services in folders.

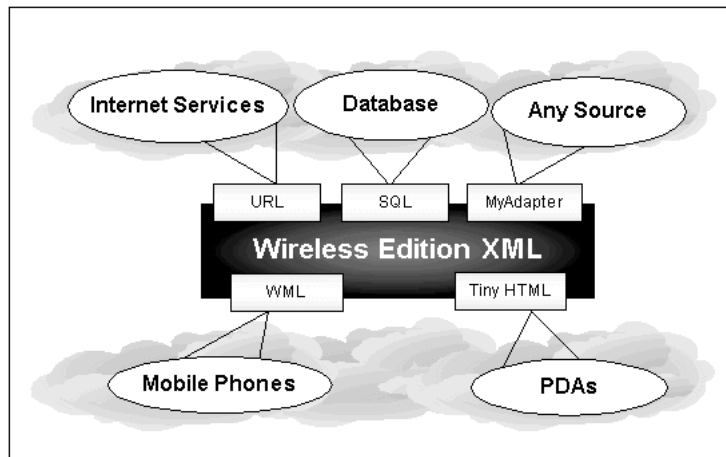
Figure 1–2 The Wireless Edition Device Portal

1.3 How Wireless Edition Works

The Wireless Edition works by isolating content acquisition from content delivery using XML (Extensible Markup Language), a standards-based language for creating structured documents. Because it isolates content from presentation, XML is an ideal language for exchanging data between diverse platforms.

The Wireless Edition provides an intermediary format layer, the Wireless Edition XML, between the source format and the target format. The Wireless Edition XML enables developers to adapt new content sources from existing target platforms, or develop new target platforms for existing content. See the *Oracle9i Application Server Wireless Edition Developer's Guide* for more information on Wireless Edition XML.

Figure 1–3 Isolating Content Acquisition from Content Delivery



Using the Wireless Edition, developers can build mobile applications that allow users to access email, calendars, flight information, or buy tickets. Users can continue to access certain applications and data when they go offline. When the applications go back online, they synchronize with backend services.

The Wireless Edition makes Web content available in a variety of markup languages. The architecture of the Wireless Edition uses services, adapters, and transformers to convert Web content into an XML object. The conversion of Web content, or source content, into an XML object is initiated when a user requests a Wireless Edition service. The Wireless Edition then processes the request as follows:

1. The request manager performs user-level preprocessing, including user authentication.
2. If the user is a valid Wireless Edition user, the request manager creates a request object, which it forwards to the master service associated with the request.
3. The master service invokes a Wireless Edition adapter to retrieve the requested content. A master service is a Wireless Edition object that implements a deliverable service. The adapter returns the content in Wireless Edition XML.

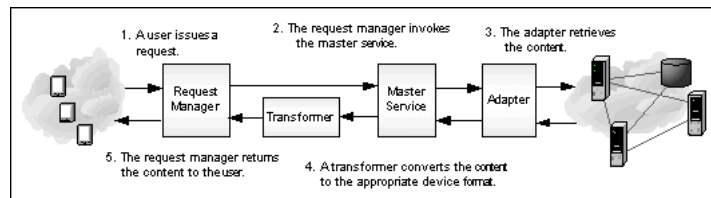
Note: A user request is not the only event that can initiate a Wireless Edition service; a scheduled job can also initiate a service.

4. A transformer, in the form of an XSLT stylesheet or Java program, converts the content into the format appropriate for the target device.

Finally, the request manager returns the information to the user.

Note: For more information on Wireless Edition adapters, see the *Oracle9i Application Server Wireless Edition Developer's Guide*.

Figure 1–4 Processing a User Request



1.3.1 Alerts

Developers can use the Wireless Edition to build notification services that alert users to important information or events. The Wireless Edition supports these notifications, or "push applications", as email, SMS, and WAP-push. Users can make their own notifications without having to access the system. In addition, developers can also build location-sensitive mobile notification services that send a mobile user personalized promotional offers from stores or restaurants within the vicinity of the mobile user.

1.4 Location-Based Services

The Wireless Edition enables developers to assign a location to a service, making the service location-based. The Wireless Edition supports two types of location-based services, location-based services that are visible to users only at certain locations and services that provide content specific to current location of the Wireless Edition user. For example, a Wireless Edition user located in San Francisco would see content related to San Francisco, such as listings of restaurants or business in San Francisco. The Wireless Edition supports content-specific location-based services through customized adapter implementation and location-aware runtime support.

The Wireless Edition provides developers with the spatial API needed to build location-based services. In addition, the Wireless Edition renders mapping services from third-party providers without Wireless Edition developers having to write custom interfaces for each service. For information on creating location-based services, see [Section 3.4, "Creating a Location-Based Service"](#) in [Chapter 3, "Wireless Edition Services"](#) and [Chapter 4, "Using the Region Modeling Tool"](#). For information on the spatial API, see the *Oracle9i Application Server Wireless Edition Developer's Guide* and the Oracle9i Application Server Wireless Edition Java documentation.

1.5 Wireless Edition Architecture

The Wireless Edition architecture is designed to transform specific content from such sources as Web sites, Internet applications, or databases into XML. The Wireless Edition core is situated between external data sources and the user's device. The core is comprised of three components that process external content by performing the following functions:

- Content Acquisition
- Content Delivery
- Core Management

1.5.1 Content Acquisition

The Wireless Edition adapters and services perform the content acquisition functions of the Wireless Edition core.

1.5.1.1 Adapters

A Wireless Edition adapter is a Java application that retrieves data from an external source and renders it in Wireless Edition XML. In the Wireless Edition architecture, adapters constitute the interface between the Wireless Edition server and the

content source. When invoked by a master service, an adapter returns an XML object that contains the service content. This XML object is processed by the core of the Wireless Edition and then delivered to the user's device as a the response to the user's service request.

Note: For more information on Wireless Edition adapters, see the *Oracle9i Application Server Wireless Edition Developer's Guide*.

The Wireless Edition provides the following pre-built adapters:

URL Adapter

The URL adapter enables integration with existing Java, JavaServer Pages (JSPs), or XML applications. The URL adapter supports parameterization and cookie management, thus providing an easy way to call Wireless Edition services from remote applications and make them wirelessly available. The URL adapter is well suited for developers wishing to build a wireless Web site or internet application using only Java servlets and JSPs.

SQL Adapter

The SQL adapter accesses any JDBC-enabled data source. Developers can use the SQL adapter to create services that query databases, invoke PL/SQL procedures, or call stored procedures applications to be deployed as wireless applications.

Servlet Adapter

The Servlet adapter enables developers to integrate other applications that are already Java servlets, providing a convenient way to call them as Wireless Edition services and make them wirelessly available.

Stripper Adapter

Given a URL, the stripper adapter dynamically retrieves and converts the content of the URL target. Unlike the Web Integration adapter, which uses a predefined mapping of the source content, the stripper adapter dynamically processes the markup tags in the content. Currently, the stripper adapter either removes the original markup tags or leaves them intact. Developers can extend the stripper adapter, however, so that it processes the tags in another way.

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 Wireless Edition 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.

The Wireless Edition provides a visual tool for creating WIDL files, the Web Integration Developer. To create a WIDL file, developers identify the elements of a Web page that they want to make accessible to a service. Developers then associate output and input parameters to the source elements that they want to access in a Wireless Edition service.

Once developers create the WIDL file, they publish it to the Web Integration Server. Publishing a WIDL file to the server makes it accessible to master services in the Wireless Edition repository. For more information on using WIDLs, see the *Web Integration Developer User's Guide* and *Web Integration Server User's Guide*.

Custom Adapter

The custom adapter is a public interface to enable development of any specific adapter for an external resource using such tools as Oracle JDeveloper. Java and XML are two custom adapter APIs. Extending the Wireless Edition adapter classes enables the custom adapter to plug into the core engine. The key requirement is the output data from the custom adapter and transformation process must be a Wireless Edition DTD-compliant XML document.

Geocoding Adapter

The Geocoding adapter provides coordinates such as longitude and latitude for a given address

Routing Adapter

The Routing adapter provides routing directions based on landmarks, the source and destination addresses, or on the geocoding of the two addresses.

Yellow-Page Adapter

The Yellow Page adapter provides the capability to locate the Yellow-Page directories and businesses based on the wireless user's current location.

Mail Adapter

The Mail adapter is based on the JavaSoft's Java Mail API and provides mail services to both POP3 and IMAP4 mail servers. It provides basic authentication and mail session management. The Mail adapter enables users to browse, create, delete, reply, and forward mail and folders from their wireless devices.

Directory Adapter

The Directory adapter provides directory service and is based on the JavaSoft Java Naming Directory Interface (JNDI) 1.2 specification. The Directory adapter also enables you to create hyperlinks.

Provisioning Adapter

The provisioning adapter enables developers to manage the Wireless Edition users from a service interface. It allows developers to add, remove, or modify users by invoking a Wireless Edition service. A developer must have administrative privileges to use the provisioning adapter.

Location-Based Service (LBS) Adapter

The location-based service (LBS) adapter provides geo-services using the physical location of the wireless device.

1.5.1.2 Services and Master Services

The Wireless Edition encapsulates external sources, such as a Web page or database information as a service. Services enable end users to access the functionality of the Wireless Edition adapters. Services represent the link between the content source and the delivery target. They tie a specific adapter to the logical devices in the Wireless Edition repository. Services also define how users access the adapter. They can restrict or grant user access to the adapter's parameters. Services can set default values for a parameter, they can enable users to set a default value, or they can hide the parameter from users altogether.

Master services provide the actual implementation of the service. They specify the adapter used for the service and any service-specific parameters. Each master service is based on one adapter. A master service creates its own instance of the adapter it uses. Therefore, several services can use the same type of adapter, and each can pass its own service-specific argument values.

A master service provides the actual implementation of the service. It specifies the adapter used for the service and any service-specific parameters

1.5.1.3 Service Manager

The Service Manager controls the services by managing the interaction between the runtime engine and a specific service. When the service manager receives a request to access a Web service, it invokes the appropriate adapter and passes the parameters required to bind to the service as bind variables and accepts the resulting XML document from the adapter. In addition, the Service Manager loads an adapter that is common to a set of services once, pools requests to access various services and manages security between the Wireless Edition and the services.

1.5.2 Content Delivery

A Wireless Edition transformer is either a Java program or an XSLT (eXtensible Stylesheet Language for Transformations) stylesheet that converts content returned by the Wireless Edition adapters into the target format of a device (for example, HDML, VoXML) or into another Wireless Edition format.

Note: For more information on Wireless Edition transformers, see the *Oracle9i Application Server Wireless Edition Developer's Guide*.

The Wireless Edition provides two types of transformers:

Generic service transformers for languages such as WML. A generic transformer can convert Wireless Edition-compliant XML into a generic WML that works on any WAP-compliant wireless phone.

Device-specific transformers are optimized for a specific device. Instead of using a generic WML transformer, a developer can use a transformer specific to a particular device that would exploit the characteristics of that device completely.

The Wireless Edition provides many pre-built adapters as well as device transformation rule files that enable developers to create support for any type of device and markup language. The transformers provided by the Wireless Edition include the following target formats:

WML 1.1

The wireless markup language defined by the WAP Forum. The Wireless Edition provides transformers for a general WML implementation, as well as for device-specific WML implementations.

Tiny HTML

A minimal HTML implementation suitable for handheld devices, such as Palm Computing Platform and Windows CE devices.

VoxML

The Motorola markup language that enables voice interaction with applications.

HDML

The Handheld Devices Markup Language is a simplified version of HTML designed specifically for handheld devices.

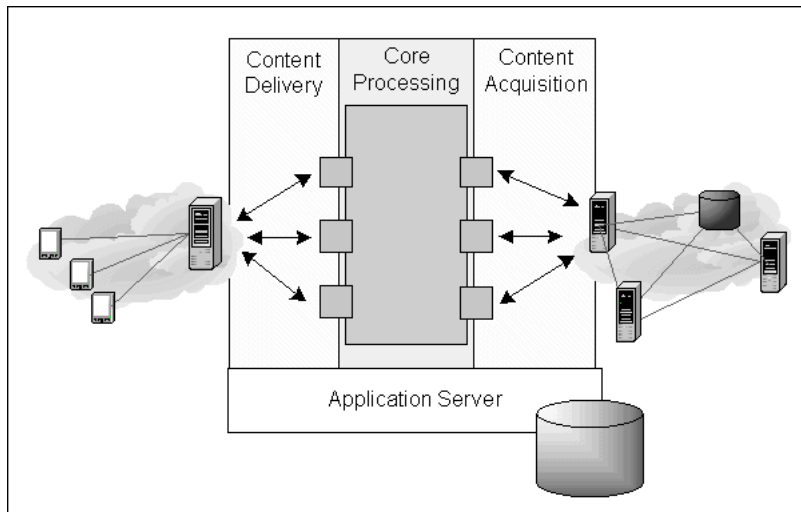
Plain Text

The plain text transformer converts content for Short Message Service-capable devices and email applications.

1.5.3 Core Processing

Wireless Edition's runtime engine abstracts the content sources as a set of wireless services that can be personalized by end users. The Wireless Edition repository stores all information about users, user preferences, and various services accessible from the Wireless Edition.

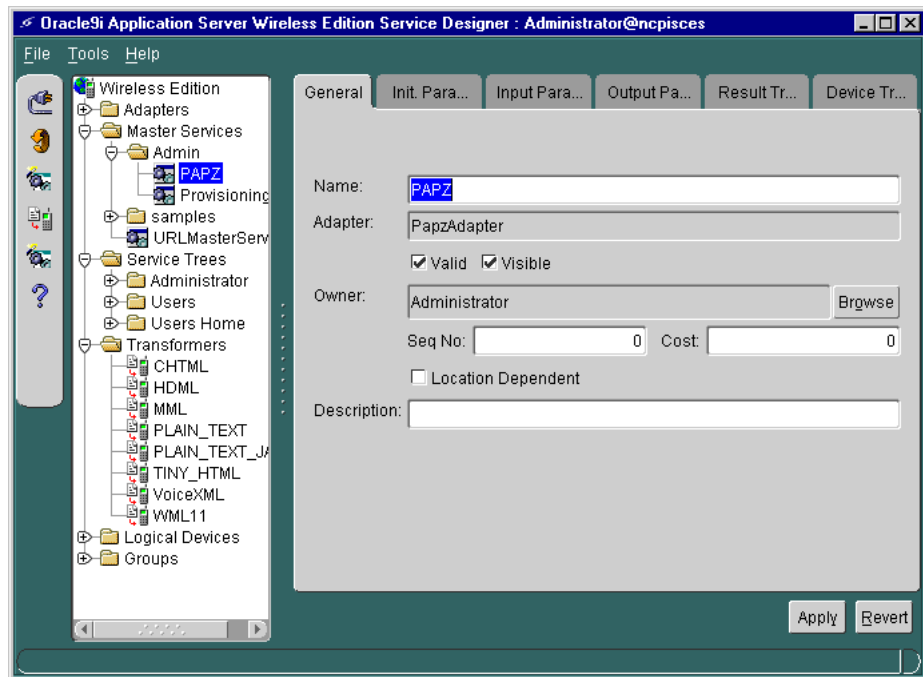
The Wireless Edition server performs the primary tasks associated with acquiring and delivering mobile services; it receives user requests, and adapts and transforms content.

Figure 1–5 The Wireless Edition Server

The Wireless Edition runs on the Oracle8i Application Server (iAS). It uses Oracle8i for its persistent object storage. When you implement the Wireless Edition, you primarily create and manage the repository objects, and establish relationships between them.

1.5.3.1 Service Designer

The Service Designer is a visual interface for implementing and managing and creating such objects in the Wireless Edition repository as services, folders, and adapters. The Service Designer includes function wizards, such as the Master Service Creation Wizard, the Transformer Testing Tool, and the Region Modeling Tool. The Service Designer also includes a pluggable API. For more information on the Service Designer, see [Chapter 3, "Wireless Edition Services"](#). For more information on the Transformer Testing Tool, see the *Oracle9i Application Server Wireless Edition Developer's Guide*.

Figure 1–6 The Service Designer

1.5.3.2 Personalization Portal

Users configure their device portals from the Wireless Edition Personalization Portal. The Personalization Portal is a Web-based interface that users access from either a desktop computer or from a mobile device.

The Personalization Portal enables users to set frequently entered parameters, such as email addresses, passwords, PINs, and account numbers.

Figure 1–7 The Personalization Portal



By modifying the elements that the Wireless Edition uses to generate the Personalization Portal, users can rebrand the Personalization Portal. The Wireless Edition uses JavaServer Pages, image files, and stylesheets to generate the Personalization Portal. For more information on configuration from the desktop, see [Chapter 6, "Personalization from Desktops"](#). For more information on personalization from a device, see [Chapter 7, "Personalization from a Device"](#).

1.6 Extensibility: Adding Adapters and Transformers

Developers can supplement the adapters and transformers that the Wireless Edition provides with those create. The Wireless Edition provides the tools and interfaces that developers can use to create customized adapters and transformers. By isolating the content source format from the target format, the Wireless Edition enables developers to easily plug in new content sources or target platforms. When

developers create an adapter for a new content source, or customize a Wireless Edition adapter, the content is immediately available to all target platforms.

Similarly, when developers create a transformer for a new device platform, all of the existing content is immediately available to the new target platform.

This provides flexibility for adapting and delivering content, but it also enables developers to present information in a device-specific manner. That is, information and applications can be customized for the form factor of each device.

Wireless Edition Tools

This document describes the tools you can use to develop and maintain the Oracle9i Application Server Wireless Edition. Each section of this document presents a different topic. These sections include:

- [Section 2.1, "Overview"](#)
- [Section 2.2, "Development Tools"](#)
- [Section 2.3, "Personalization"](#)

2.1 Overview

The Wireless Edition provides tools to help you create, manage, and deliver mobile services. These tools include visual interfaces for developing and managing repository objects, and utilities for managing the server and deploying the Wireless Edition.

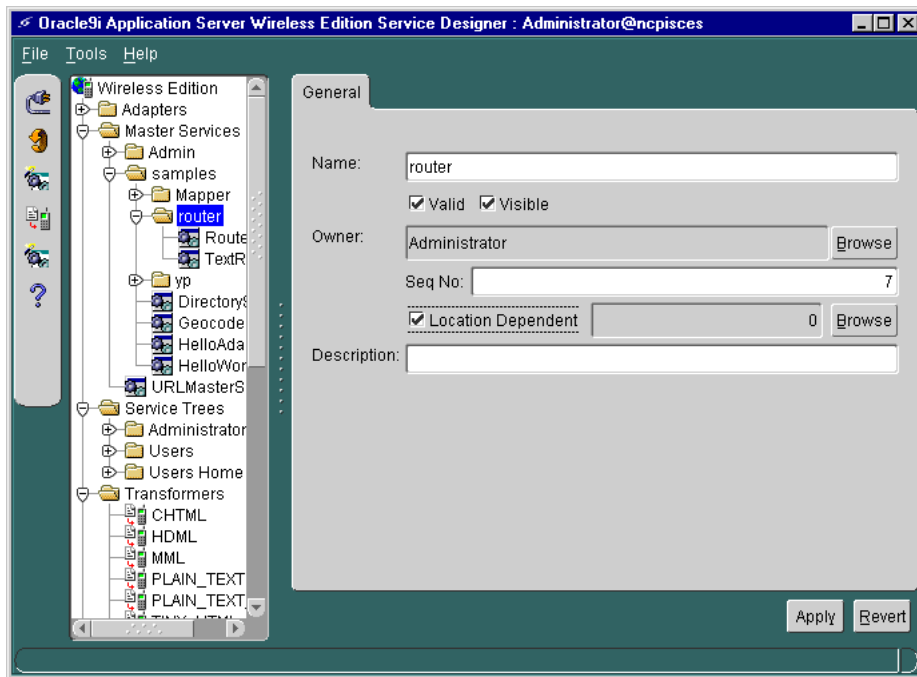
2.2 Development Tools

The Wireless Edition provides the following visual development tools:

- Service Designer
- Master Service Creation Wizard
- Region Modeling Tool

2.2.1 Service Designer

The Service Designer is a visual interface for implementing and managing the Wireless Edition.

Figure 2–1 The Service Designer

You use the Service Designer to create and modify objects in the Wireless Edition repository. These objects include:

Table 2–1 Objects in the Wireless Edition Repository

Object Type	Description
Adapter	Adapter objects represent the Wireless Edition interface to content sources. Adapter objects have an attribute called <code>classes</code> . This attribute identifies the archive file that contains the actual Java implementation of the adapter.
Master service	The master service object implements the service and invokes a specific adapter. You can associate a custom transformer with a master service. Custom transformers are service- and device-specific. They can be used by only one device and one master service.

Object Type	Description
Alias	An alias is a pointer to a master service, folder, bookmark, or other alias. You can use aliases to distribute access to services and to override default parameter values defined by the master service. You can make a service accessible to a user by placing an alias to the service in a tree the user owns or can access.
Folder	You can use folders to group services. Folders make services available to end users. Every user has a "home" folder. This contains the services, usually referenced by an alias, that the user can access. A user can also access any service in a folder owned by a group to which the user belongs.
User group	User groups help you manage service access for multiple users. You can create user groups based on factors such as subscription level, geographic location, or interests.
Transformer	<p>A transformer converts the content returned by the Wireless Edition adapters. Transformer types include:</p> <ul style="list-style-type: none"> ■ Result transformers, which convert Adapter Result content into SimpleResult content. ■ Device transformers, which convert SimpleResult content into the final target format. <p>A device transformer can be either the default transformer for a logical device, or a custom transformer, which is used to render a specific master service for a specific logical device.</p>
Logical device	A logical device object associates a physical device or an abstract device, such as email, with a transformer.
Bookmarks	End users can set bookmarks to external URLs at the Personalization Portal. The bookmark appears as a menu selection in the device portal. The Wireless Edition does not process the content of the URL target. The format of the target content must be supported by the user's device.

The Service Designer provides a tree view of the Wireless Edition repository. The tree shows Wireless Edition object classes, such as adapters and transformers, as folders, or branch nodes. For more information on using the Service Designer, see [Chapter 3, "Wireless Edition Services"](#). See the *Oracle9i Application Server Wireless Edition Developer's Guide* for more information on transformers and adapters.

The Service Designer includes the following tools:

- Master Service Creation Wizard

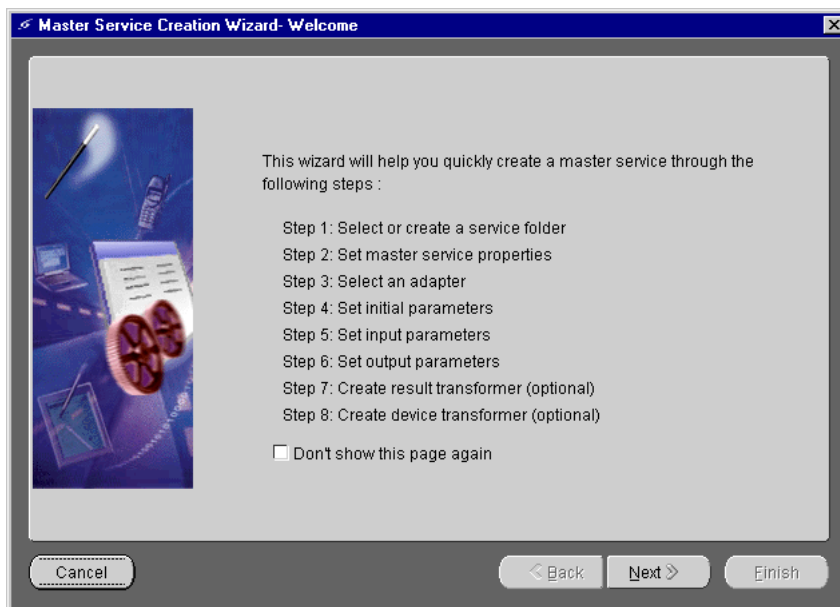
- Transformer Creation and Testing Tool
- Region Modeling Tool

These tools, which you access from the Tools menu in the Service Designer, enable you perform administrative tasks quickly and easily.

2.2.1.1 Master Service Creation Wizard

In addition to the Service Designer, The Wireless Edition enables you to successfully create a service with the Master Service Creation Wizard. The Master Service Creation Wizard guides you through the creation of a master service by presenting a separate screen for each step in the sequence. For more information using the Master Service Creation Wizard, see [Section 3.3, "Creating a Master Service"](#) in [Chapter 3, "Wireless Edition Services"](#) and [Chapter 9, "Walkthroughs"](#).

Figure 2–2 *The Welcome Screen of the Master Service Creation Wizard*



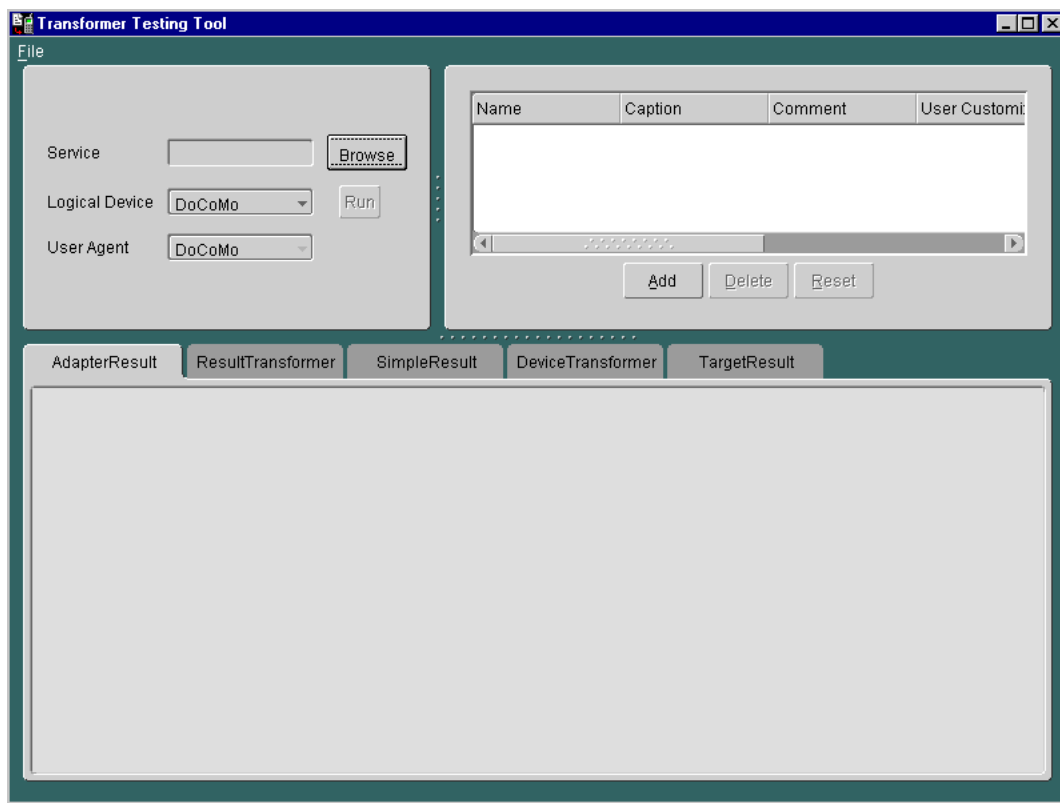
2.2.2 The Transformer Testing Tool

The Wireless Edition includes the Transformer Testing Tool, a tool for testing and creating transformers. Transformers, in the form of XSLT stylesheets or Java classes, convert the content returned by the Wireless Edition adapters into the format best

suited to a particular platform. See the *Oracle9i Application Server Wireless Edition Developer's Guide* for more information on Wireless Edition XML.

The Transformer Testing Tool enables you to both test and create transformers using a single screen. The tool allows you to view and edit adapter input parameters and the transformers of the source content. For more information on using the Transformer Testing Tool, see the *Oracle9i Application Server Wireless Edition Developer's Guide*.

Figure 2–3 *The Transformer Testing Tool*

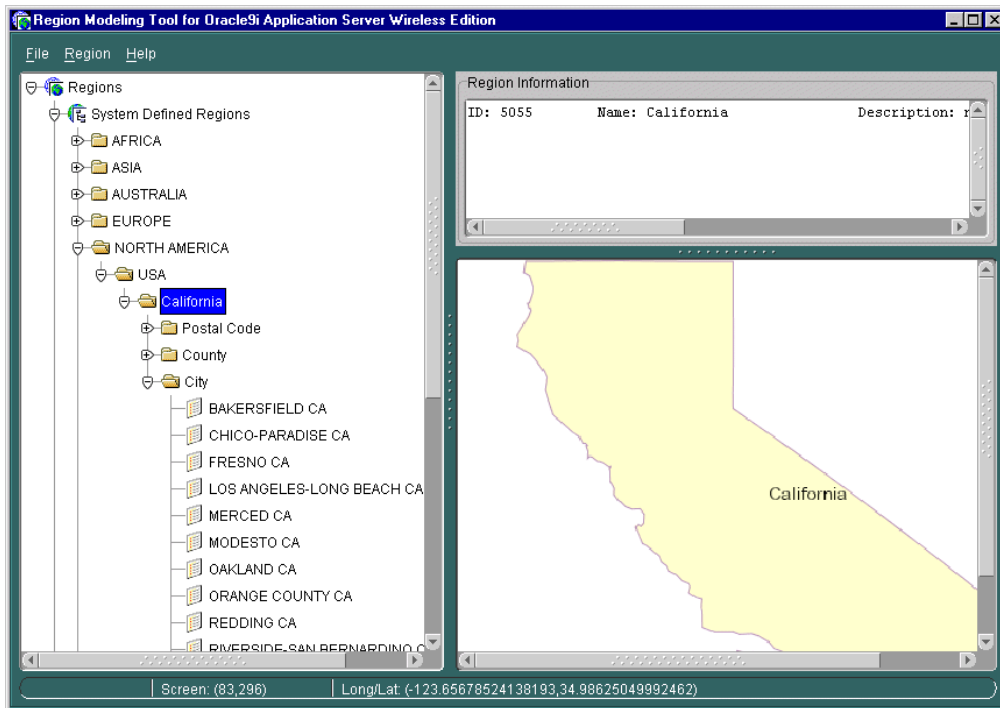


2.2.3 Region Modeling Tool

The Wireless Edition enables you to create location-based services using the Region Modeling Tool. Using this graphical interface, you can create services that are visible to Wireless Edition users at specific locations.

The Region Modeling Tool enables developers to create these location-based services by assigning a location to a folder or to a master service from the spatial data repository. The tool enables you to view a spacial object as a map image with its corresponding geometry. In addition, the tool enables you to access, view, and modify the data stored in the spatial database in the Wireless Edition repository. For more information on the Region Modeling Tool, see [Chapter 4, "Using the Region Modeling Tool"](#).

Figure 2–4 The Region Modeling Tool



2.2.4 The Web Integration Developer

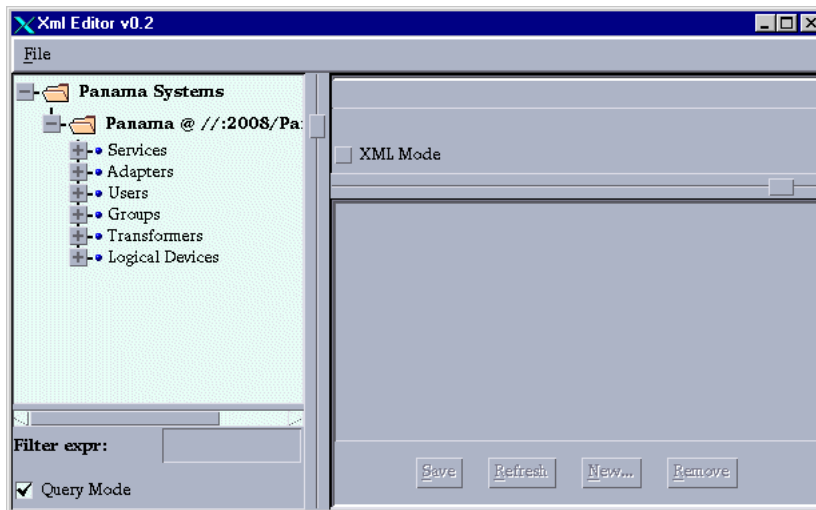
The Web Integration Developer is a visual interface for creating and testing WIDL services. The Web Integration adapter uses WIDL services to map Web content. You also use the Web Integration Developer to publish the WIDL interfaces to the Web Integration Server.

When you create a WIDL service, you parameterize a source page. You create input parameters for input elements, such as text fields and selection lists, and output parameters for returned content. The Web Integration adapter renders the Web content in one of the Wireless Edition result formats. It converts the content according to the rules defined in the adapter. For example, record variables, which you can create in the Web Integration Developer, are data structures that assemble related variables in a group. The Web Integration adapter converts record variables into menu items, and assumes that the first element in a record is a linked name and the second is the target URL.

For more information on creating WIDL services and interfaces, see *Oracle9i Application Server Wireless Edition Developer's Guide* and [Section 9.8, "Creating a Master Service Using the Web Integration Adapter"](#) in [Chapter 9, "Walkthroughs"](#).

2.2.5 XML Editor

The Wireless Edition XML Editor is a text editor that allows you to create, modify, and delete repository objects as XML elements. You should not attempt to use the XML Editor unless you understand the XML structure of the Wireless Edition repository. Wireless Edition does not validate the changes you make with the XML Editor. You should always back up your repository before modifying it with the XML Editor.

Figure 2–5 The XML Editor

For more information on using the XML Editor, see the *Oracle9i Application Server Wireless Edition Developer's Guide*.

2.2.6 Oracle XML Parser

The Wireless Edition processes XML documents using the Oracle XML Version 2 parser. The parser supports the Document Object Model (DOM) and the Simple API for XML (SAX) interfaces. It supplements the DOM interface with extensions, which are detailed in the *Oracle9i Application Server Wireless Edition API Specification*.

2.3 Personalization

Wireless Edition enables users to personalize their portals from a desktop computer or from a variety of wireless devices.

2.3.1 Personalization from Desktop

The Personalization Portal enables users to personalize their portals from a desktop computer by providing default pages for modifying service, folder, job, user, group and other Wireless Edition repository objects. Users can also create their own customized pages, or they can add pages to update customer data objects. The framework and interface are designed to be extensible for including display/action

JSP pages specified in the **Portal.properties** file or in the Wireless Edition repository. For more information, see [Chapter 6, "Personalization from Desktops"](#).

Figure 2–6 *The Personalization Portal*

The screenshot shows the Oracle Application Server Wireless Edition Personalization Portal. At the top left is the Oracle logo and the text "Application Server Wireless Edition". To the right are "SignIn" and "Help" links. Below the header is a "Please log on" prompt with a key icon. A link for "New to our portal? Click here to register!" is present. The main content area is a login form titled "Enter username & password" with fields for "Username" (containing "Administrator") and "Password" (containing "*****"), a "Login" button, and a link for "Forgot password? Try password hint".

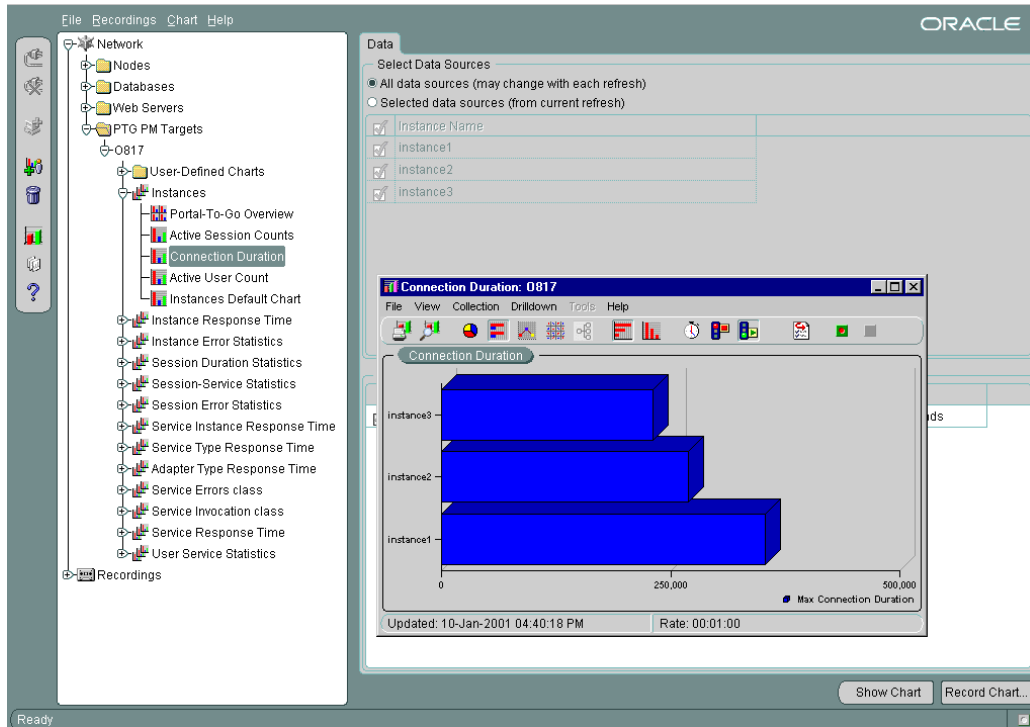
2.3.2 Personalization from Device

Wireless Edition users can also personalize their portals from a variety of wireless devices. For more information, see [Chapter 7, "Personalization from a Device"](#).

2.4 Oracle Performance Manager

The Oracle Performance Manager enables you to view performance data generated by the Data Gatherer running against the Wireless Edition repository. At runtime, the Wireless Edition servers log performance metrics into the Wireless Edition repository and each Wireless Edition instance logs statistical information. The Data Gatherer then synthesizes this information and displays it as graphs and charts.

Figure 2-7 The Oracle Performance Manager



2.4.1 Accessing the Oracle Performance Manager

To access the Oracle Performance Manager:

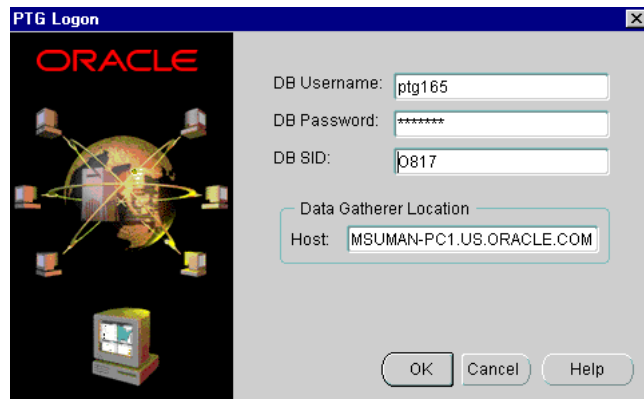
1. From Programs in Windows NT, select Oracle for Windows NT.
2. Point to Diagnostics Pack.
3. Select Performance Manager. The Oracle Performance Manager appears.

2.4.2 Logging into the Oracle Performance Manager

To log into the Oracle Performance Manager:

1. From the Oracle Enterprise Manager Navigator Tree in the left frame of the screen, select a service. For example, 0817. The Login Screen appears.
2. Select Standalone Login.

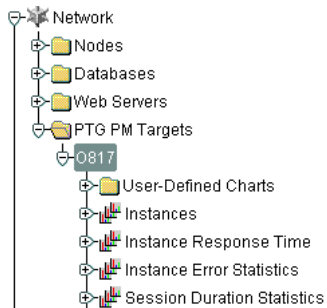
Figure 2–8 The Login Screen of the Oracle Performance Manager



3. In the login screen, enter the database user name and password and the database service ID number (for example, 0817). These are the same as the user name and password for the Wireless Edition System Logger.
4. Enter the host name of the machine where the Data Gatherer is installed (for example, `mymachine.us.oracle.com`). The Data Gatherer must in the same host as the Oracle Instance.

Note: The Oracle Performance Manger uses SQL*Net to connect to the repository. Therefore, `tnsnames.ora` must be configured properly.

After you have logged in successfully, a list of statistics that have been collected by the Wireless Edition Data Gatherer cartridge displays in the Navigator Tree under PTG Performance Manager Targets.

Figure 2–9 The Navigator Tree

Each statistic displays a table chart, or as a default bar chart. The data is grouped by their Wireless Edition Instance IDs.

2.4.3 Performance Data Collection Schema

The Oracle Performance Manger logs performance data in the following tables.

Note: The runtime API logs the information into the tables. The collection cartridges retrieve this information and deliver it to the Oracle Enterprise Performance Manager and Capacity Planner.

2.4.3.1 PTG_SERVICE_LOG

Statistical information is logged in this table when responding to a service request with consolidated service information. The request_status is 0 (for 'SUCCESS') by default. In case of an error, the status is set to 'NON-ZERO' and the corresponding error description is given.

Table 2–2 PTG_SERVICE_LOG

Column Name	Type	Comments
Service_id	Number	Notnull
Service_name	varchar2(80)	Notnull
ptg_instance_id	varchar2(240)	Notnull
Final_Service_id	Number	Notnull
Final_Service_Name	varchar2(80)	Notnull

Column Name	Type	Comments
Session_id	Number	NotNull
Service_arguments	varchar2(240)	(can turn on/off)
Bookmark	varchar2(80)	
Service_type	varchar2(80)	
Invocation_hour	Number	NotNull-values 0-23
Invocation_time	Number	NotNull
Response_time	Number	NotNull default 0
request_status	Number	Default 0 for success
Error_description	varchar2(4000)	
Userid	number	NotNull
Username	varchar2(80)	NotNull
External_Userid	number	NotNull
External_Username	varchar2(80)	NotNull
Remote_address	varchar2(240)	NotNull
Host_id	varchar2(240)	NotNull
Transformation_time	number	
Logical_device	varchar2(80)	NotNull

Views provide the pertinent statistical information from the PTG_SERVICE_LOG and the PTG_SESSION_LOG.

2.4.3.2 PTG_SESSION_LOG

This log is created at login time and updated at logout time or at session expiry.

Table 2-3 PTG_SESSION_LOG

Column Name	Type	Comments
Session_id	Number	NotNull
User_name	varchar2(80)	NotNull
User_id	number	NotNull

Column Name	Type	Comments
Ptg_instance-id	varchar2(240)	Notnull
Remote_address	number	Notnull
Logical_device	varchar2(240)	Notnull
Login_time	date	Notnull
login_hour	number	Notnull-values 0-23
Expiry_Time	date	<updated at logout time>

2.4.3.3 SERVICE_REQUEST_STATS

This view has records within a particular window (i.e., records where the service log timestamp is within a given window).

Table 2-4 SERVICE_REQUEST_STATS

Column Name	Type	Comments
Service_id	Number	Notnull
Service_name	varchar2(80)	Notnull
Service_type	varchar2(80)	Notnull
Ptg_instance-id	varchar2(240)	Notnull
Invocation_count	number	Notnull
Avg_response_time	number	Notnull <Threshold defined on this>
Ptg_instance_id	varchar2(240)	

2.4.3.4 SERVICE_ERROR_STATS

Table 2-5 SERVICE_ERROR_STATS

Column Name	Type	Comments
Service_id	Number	Notnull
Service_name	varchar2(80)	Notnull
Service_type	varchar2(80)	Notnull
Total_Error_count	varchar2(80)	<Threshold defined on this>

Column Name	Type	Comments
Ptg_instance_id	varchar2(240)	

2.4.4 Performance Statistics

Oracle Performance Manager console uses the following statistical methods to measure system events. Each of these methods has a corresponding folder in the Oracle Enterprise Manager Navigator tree displayed in the left frame of the Oracle Performance Manager screen. Selecting one of these folders (for example, Connection Duration), displays the statistics as a graph in the right frame of the screen.

2.4.4.1 Session Duration Statistics

Session duration statistics are a class of statistics that present the duration of each session. The data is grouped by instance IDs. The duration of each session is computed using the login time and the expiry time (or the current time if the session is still operational). The Oracle Performance Manager presents session duration statistics either as a pie chart or as a table chart.

2.4.4.2 Session Error Statistics

Session error statistics are a class of statistics that represent the number of errors for each session. For session duration statistics, the data is grouped by instance IDs.

2.4.4.3 Session Service Statistics

Session service statistics is a class of statistics that represent the number of services invoked during each session. The data is grouped by instance IDs.

2.4.4.4 Service Error Statistics

Service error statistics are a class of statistics that represent the number of services which have runtime errors. The service error data is grouped by instance IDs. When the number of service errors exceeds a configurable threshold, the Oracle Performance Manager generates a warning and an error event on the Oracle Enterprise Manager Console. The warning and error threshold values are configurable from the Oracle Enterprise Manager Console.

2.4.4.5 User Service Statistics

User service statistics are a class of statistics that presents the number of services invoked by a specific user across instances. The user service statistics data is grouped by user name and by instance IDs.

2.4.4.6 Service Invocation Statistics

Service invocation statistics are a class of statistics that represent the number of times a particular service was invoked across instances. The service invocation statistics data is grouped by service names and instance IDs.

2.4.4.7 Service Response Time Statistics

The service response time statistics are a class of statistics that present the average response times for each service that was invoked across instances. The service response time statistics are grouped by service names and instance IDs. If the response time exceeds a configurable threshold value, then the Oracle Performance Manager generates a warning or an error even onto the Oracle Enterprise Manager Console.

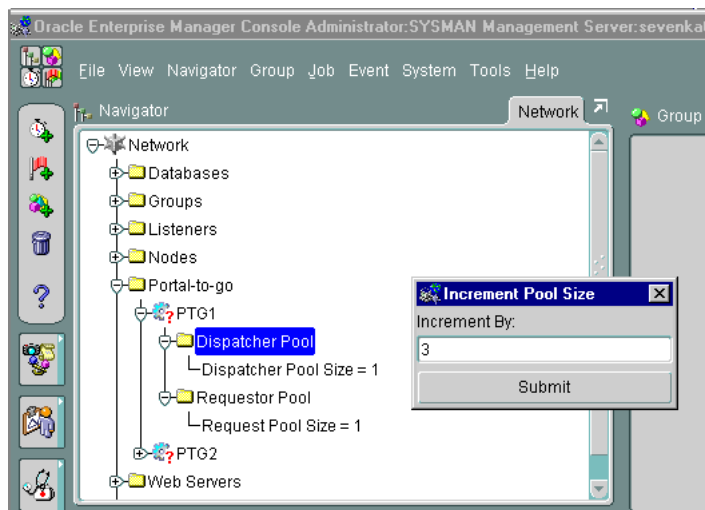
2.4.5 Fault Management

The event framework of the Oracle Enterprise Manager enables asynchronous notification of error information. When an error occurs, the Wireless Edition generates an Oracle Enterprise Manager event at the node where the error occurred and provides the details necessary for fault diagnosis. The Intelligent Agent running on each node picks up these Oracle Enterprise Manager events and forwards them to the Oracle Enterprise Manager Console, where they are immediately displayed. You can configure the console to notify the administrator of an error using pages or email.

2.4.6 Configuration Management

The Wireless Edition provides a framework for Wireless Edition discovery on the console. When you discover this node on the Oracle Enterprise Manager Console, you can use the console to both monitor runtime objects and to perform limited performance tuning.

Each Wireless Edition instance starts an RMI server, which listens to commands sent from the RMI client running on the client. The Oracle Enterprise Manager Navigator Tree enables real-time control of thread pool, connection pool, status query, and update.

Figure 2–10 Using the Oracle Enterprise Manager Console to Increment Pool Size

In the above figure, the Oracle Enterprise Manager Console displays a node with an instance name of PTG1. The controls for PTG1 display under the PTG1 folder. You can modify these controls through a right-click. For example, clicking the Dispatcher Pool control invokes the option to increment the pool size.

For more information on configuration, see the Oracle9i Application Server Wireless Edition Configuration Guide.

Wireless Edition Services

This document describes how to create and manage the services of the Oracle9i Application Server Wireless Edition using the Service Designer. Each section of this document presents a different topic. These sections include:

- [Section 3.1, "Overview"](#)
- [Section 3.2, "Using the Service Designer"](#)
- [Section 3.3, "Creating a Master Service"](#)
- [Section 3.4, "Creating a Location-Based Service"](#)
- [Section 3.5, "Modifying a Master Service"](#)
- [Section 3.6, "Deleting a Master Service"](#)
- [Section 3.7, "Creating a Folder"](#)
- [Section 3.8, "Deleting a Folder"](#)
- [Section 3.9, "Creating a Service Alias"](#)

3.1 Overview

Services enable end users to access the functionality of the Wireless Edition adapters. Services represent the link between the content source and the delivery target. They tie a specific adapter to the logical devices in the Wireless Edition repository.

Services also define how users access the adapter. They can restrict or grant user access to the adapter's parameters. Services can set default values for a parameter, they can enable users to set a default, or they can hide the parameter from users altogether.

The following repository objects encapsulate and manage the Wireless Edition services:

- Master services
- Aliases
- Folders (or service trees)

Master Services

Master services provide the actual implementation of the service. They specify the adapter used for the service and any service-specific parameters. By mapping an adapter to device transformers, master services link the Wireless Edition content sources to the delivery platforms. Each master service is based on one adapter. A master service creates its own instance of the adapter it uses. Therefore, several services can use the same type of adapter, and each can pass its own service-specific argument values.

The Wireless Edition end users typically invoke a master service by clicking on a menu item from their device. The information returned by the master service can be text, such as a movie review, or an application, such as an airline booking system. While a user request usually invokes a master service, a scheduled job can also invoke a master service.

Location-Based Services

The Wireless Edition enables developers to assign a location to a service, making the service location-based. The Wireless Edition supports two types of location-based services, location-based services that are visible to users only at certain locations and services whose content depends upon the physical location of the user. For example, Wireless Edition users located in San Francisco would see content related to San Francisco, such as lists of restaurants or business located in San Francisco. The Wireless Edition supports content-specific location-based services through customized adapter implementation and location-aware runtime support. For more information on creating location-based services that are visible only at specific locations, see [Section 3.4.1](#) and [Section 4.4](#), "Creating Location-Based Master Services and Folders" in [Chapter 4](#), "Using the Region Modeling Tool". For information on creating location-based services that display content specific to the user's location, see [Section 4.5](#), "Creating Location-Based Services with Location-Specific Content" in [Chapter 4](#), "Using the Region Modeling Tool".

Aliases

An alias is a pointer to a master service, folder, bookmark, or to another alias. Aliases are useful for distributing access to a master service among many users and groups. End users can access a master service only if the master service, or an alias to the master service, appears in a service tree they own or that a group they are a member of owns. Aliases can set parameter values that override values set at the master service.

Folders

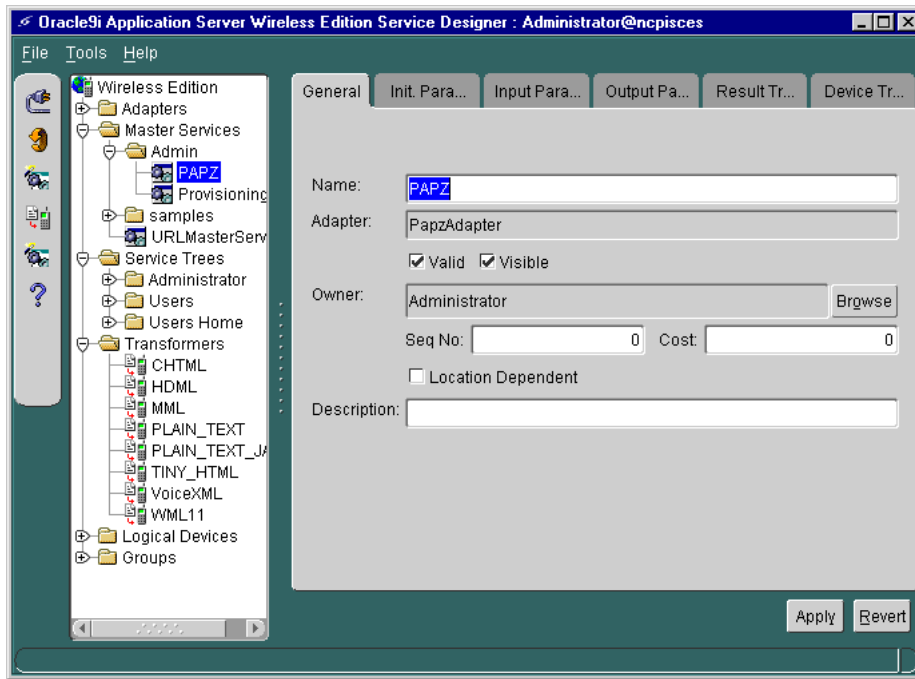
Folders group services and make them available to end users. Every user has a "home" folder. This folder contains the services, usually referenced by an alias, that the user can access. A user can also access any service in a folder owned by the group to which the user belongs.

You can manage service objects (master services, aliases, and folders) using the Service Designer. The Service Designer provides a tree view of the Wireless Edition repository. The repository tree includes a Master Services folder and a Service Trees folder for managing services. You can create service objects in either folder. In most cases, however, you should place folders and master services in the Master Services root folder, and user- and group-owned folders and aliases in the Service Trees root folder.

3.2 Using the Service Designer

The Service Designer is a visual interface for implementing and managing the objects in the Wireless Edition. The Service Designer enables you to create and modify Wireless Edition objects, including adapters, transformers, and services.

Figure 3–1 The Service Designer



The Service Designer provides a tree view of the Wireless Edition repository. The tree shows Wireless Edition object classes, such as adapters and transformers, as folders or branch nodes. It shows instances of those classes, such as the WML 1.1 transformer, as objects, or leaf nodes.

Note: For performance reasons, the default tree view in the Service Designer does not display more than 100 objects of any type. This, however, can be configured in the **ptgsd.properties** file.

3.2.1 Starting the Service Designer

To start the Wireless Edition Service Designer:

1. From the Start menu, point to the Oracle for Windows NT menu item.
2. Click Wireless Edition, then Service Designer. The Log In dialog appears. Complete the Log In dialog as follows:

- a. Enter your user name (for example, Administrator)
- b. Enter your password (for example, manager)
- c. In the location field, specify the URI (Uniform Resource Identifier) of the repository that you want to develop. You can connect to the Wireless Edition server using RMI. The value you enter in the Location field determines the connection mode.

Note: You cannot connect directly to the Wireless Edition repository using in-process connection. You should use an RMI connection.

3.2.1.1 RMI Connection

To connect to the repository through the Wireless Edition server, the RMI listener must be running on the Wireless Edition server. By default, the RMI listener is started when you start the Wireless Edition server.

In the Location field of the Log In dialog, enter the URI of the Wireless Edition server in any of the following formats:

- *rmi://hostname:port/servername*
- *rmi://hostname:port*
- *rmi://hostname*
- *rmi://:port/servername*
- *rmi://:port*
- *//hostname:port/servername*
- *//hostname:port*
- *//hostname*
- *//:port/servername*
- *//:port*
- *hostname:port/servername*
- *hostname:port*
- *hostname*

- `:port/servername`

- `:port`

The default port number and server name are:

- Port: 2008
- Server name: PanamaServer

You can modify these defaults in the **oracle.panama.core.admin.Rmi.properties** file. The **System.properties** file identifies the default RMI listener used as follows:

```
locator.request.daemon.classes=oracle.panama.core.rmi.server.ServerImpl
```

3.2.2 Creating Objects

You can create an object in the Service Designer as follows:

1. In the repository tree, highlight the master service folder.

Note: Before using the Service Designer to create a master service, you must first use the Web Integration Server to publish the service interface to the repository tree.

2. Right-click the class or folder. A pop-up menu appears with menu items for creating objects.
3. Click the appropriate menu item for creating the new object.

The Service Designer then presents a form, or a sequence of forms, that lets you configure the object. You can navigate through the forms by clicking the Next or Previous button. When you have finished configuring the master service, click the Finish button.

3.2.3 Modifying Objects

When you highlight an object in the repository tree, the right panel displays the object's properties. If an object has more than one property panel, you can navigate between panels by clicking the tabs at the top of the panel.

You can modify an object by making changes directly to the properties in the panel. To save your changes, click the Apply button. This saves the changes you have made to any of the object's panels, not just the current panel.

To cancel the changes you have made to an object since last saving, click the Revert button. This restores the object to its prior state. If you make changes to an object, but click another object in the repository before clicking Apply, the changes to the object do not take effect.

3.2.4 Deleting Objects

To remove an object from the repository:

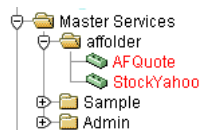
1. Highlight the object in the tree view and right-click.
2. In the pop-up menu, select the menu item for deleting the object.
3. Confirm the action.

You cannot delete an adapter used by a master service. If you delete a master service that is referenced by aliases, Service Designer deletes the aliases as well.

3.2.5 Flagged Objects

Flagged objects appear in red type in the repository tree. The Wireless Edition flags objects that are specified as not valid or not visible. Specifying an object as not valid or not visible prevents its use. You may choose to disable an object, for example, while it is under development or testing.

Figure 3–2 *Flagged Objects in the Repository Tree*



3.2.6 Refreshing the Repository View

The Service Designer caches object information. If multiple Service Designer users work in the same repository concurrently, the object view for each user may not be up-to-date with the state of the repository. To retrieve the latest state of the repository, click the refresh icon on the toolbar.

3.2.7 Object Identifiers

Each object in the Wireless Edition repository is identified by a unique object ID. You can see the object ID for an object by moving the mouse over the object. The object ID helps you to identify the object when you access the repository with another tool, such as the Wireless Edition XML Editor.

3.3 Creating a Master Service

The Service Designer provides you with two methods of creating a master service in the Wireless Edition repository: you can use the Service Designer itself as described in [Section 3.2.2, "Creating Objects"](#), or through the Master Service Creation Wizard. Using the Wizard, you can successfully create a master service by following a series of steps. The Wizard provides you with a separate screen for each step, presenting the creation of a master service as a sequence. When you complete this sequence, you complete a master service. This document describes devotes a section for each step of the master service creation sequence. These sections include:

- [Section 3.3.1, "Step 1: Selecting or Creating a Service Folder"](#)
- [Section 3.3.2, "Step 2: Setting the Master Service Properties"](#)
- [Section 3.3.3, "Step 3: Selecting an Adapter"](#)
- [Section 3.3.4, "Step 4: Setting the Init Parameters"](#)
- [Section 3.3.5, "Step 5: Setting the Input Parameters"](#)
- [Section 3.3.6, "Step 6: Setting the Output Parameters"](#)
- [Section 3.3.7, "Step 7: Creating the Result Transformer \(Optional\)"](#)
- [Section 3.3.8, "Step 8: Setting the Device Transformer \(Optional\)"](#)

The Cancel, Help, Back, and Next buttons appear at the bottom of each of the Service Creation Wizards screens. The Back and Next buttons navigate you through the service creation sequence. Clicking the Next button takes you to the next screen, clicking the Back button takes you to the previous screen. Selecting the Cancel button closes the Master Service Creation Wizard and sets all conditions back to their original state.

After you select the Master Service Creation Wizard option from the Tools menu of the Service Designer, the Welcome screen appears. Click Next to move to the Select Service Folder screen.

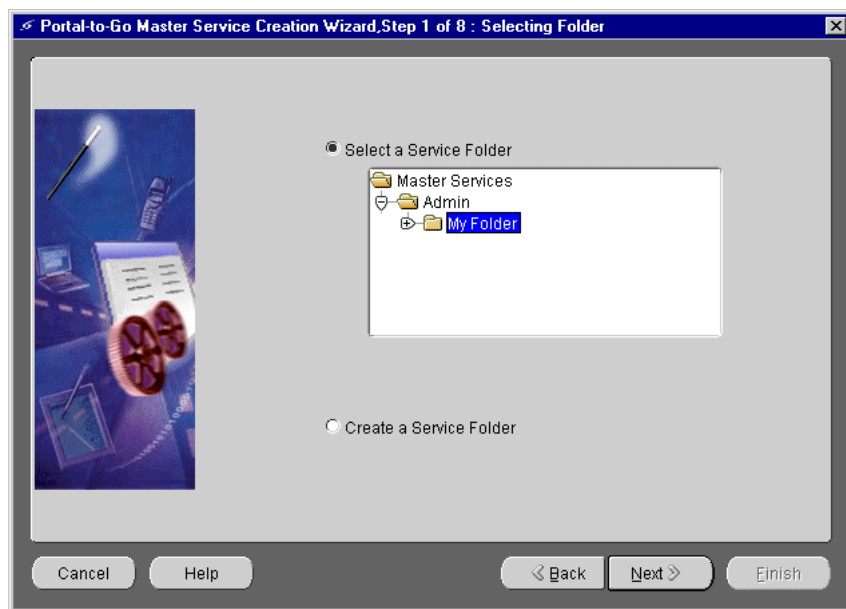
3.3.1 Step 1: Selecting or Creating a Service Folder

To select a service folder, click the radio button for **Select a Service Folder**. Click the desired folder from the tree view and then click **Next**.

Creating a New Service Folder

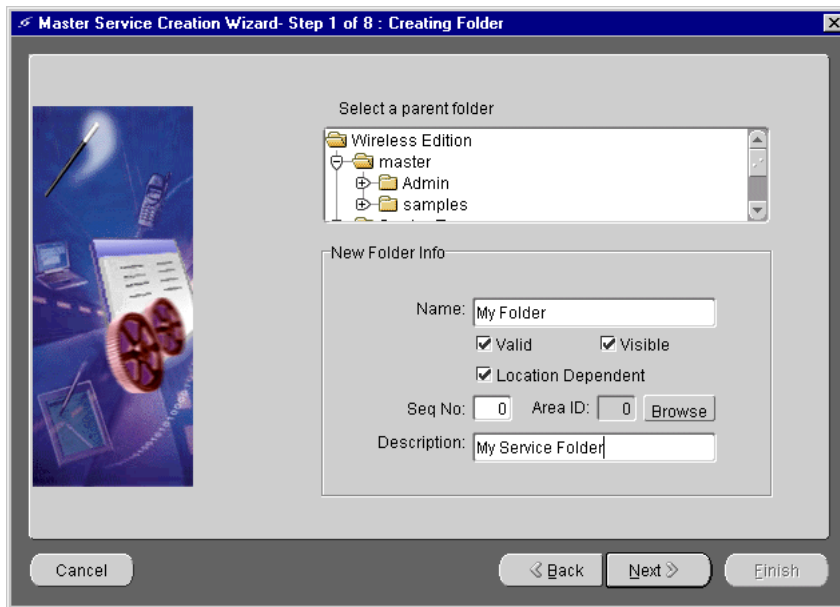
If you wish to create a new service folder, click the *Create a Service Folder* radio button and then click **Next**.

Figure 3–3 The Select Folder Screen



The Create New Folder screen appears, which prompts you to select a parent folder and then create a new service folder as a subfolder of that parent folder.

Figure 3–4 The Create New Folder Screen



The Create New Service Folder screen enables you to configure the new service folder by entering values for the following parameters:

Table 3–1 Parameters for the Create New Service Folder Screen

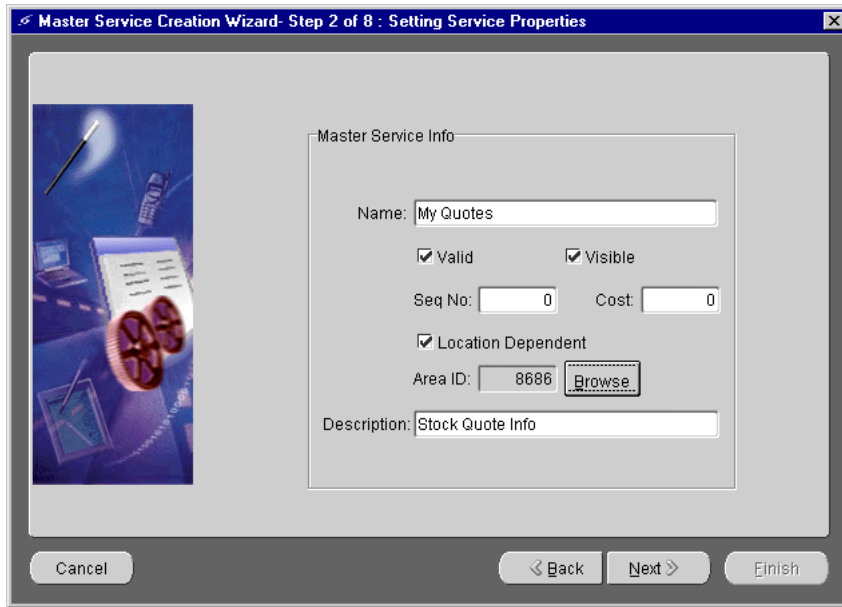
Parameter	Value
Name	The name of the service folder. The service folder name must be a unique name in the parent folder.
Valid	Select the Valid check box to enable the service folder.
Visible	Select the Visible check box to make the service folder accessible to users.
SeqNo	The integer value that you enter in this field lets you alter the order in which services and folders appear on output devices. By default, these appear in order by sequence number, then by name. You can enter values in the sequence fields to rearrange the order in which the services and folders appear.
Browse	Clicking this button invokes the Region Modeling Tool. The Region Modeling tool enables you to assign a region to a service folder, making the folder location-based.

Parameter	Value
AreaID	<p>The integer value retrieved by the browse function allows you to specify whether a service can be made visible based on a user's physical location. This enables you to implement a location-specific service. This is an optional value.</p> <p>To use this parameter, you implement the <code>requestBegin()</code> method in the <code>ParmHook</code> (request manager) interface. <code>requestBegin()</code> is called every time that the Wireless Edition receives a user request. You can place logic in this method that invokes an external position system. <code>requestBegin()</code> then passes the result in the request object. When the Wireless Edition invokes the service, it calls <code>isRuntimeVisible()</code>, passing it the user's position. You can test the position against the Area ID configured for the master service in <code>isRuntimeVisible()</code> to determine whether to make the service available.</p>
Description	An optional description of the service folder.

Click Next after you have entered the desired values. The Selecting Folder screen reappears with the Select a Service Folder radio button selected (the default setting). Select the folder you created from the hierarchy and then click Next. The Setting Service Properties screen appears.

Note: The Master Service Creation Wizard prevents you from proceeding if you enter an invalid value.

Figure 3–5 The Setting Service Properties Screen



3.3.2 Step 2: Setting the Master Service Properties

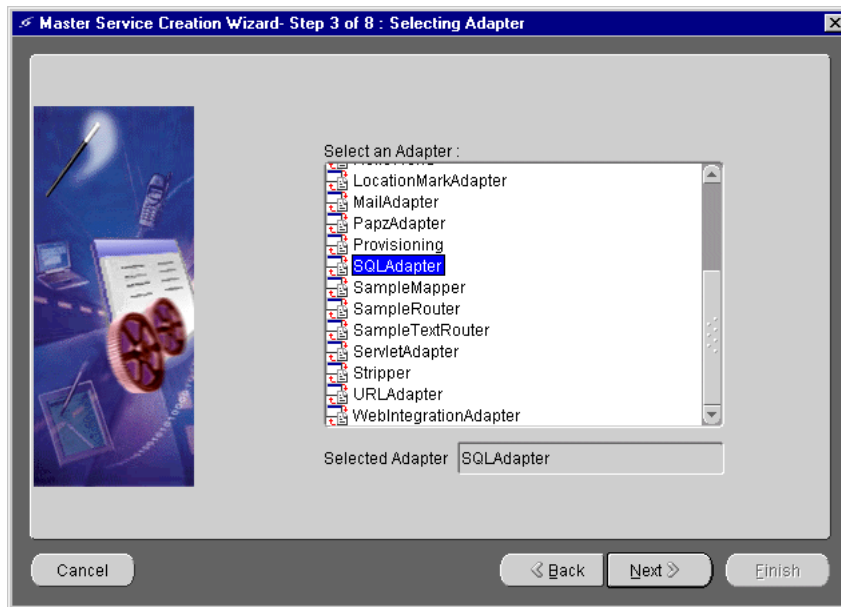
The Setting Service Properties screen enables you to configure properties for the master service by entering values for the following parameters:

Table 3–2 Configuration Properties for the Master Service

Parameter	Value
Name	The name of the master service.
Valid	Select the Valid check box to enable the master service.
Visible	Select the Visible check box to make the master service accessible to users.
SeqNo	The integer value that you enter in this field lets you alter the order in which services and folders appear on output devices. By default, these appear in order by sequence number, then by name. You can enter values in the sequence fields to rearrange the order in which the services and folders appear.

Parameter	Value
Browse	Clicking this button invokes the Region Modeling Tool. The Region Modeling tool enables you to assign a region to a service folder, making the folder location-based.
AreaID	<p>The integer value retrieved by the browse function allows you to specify whether a service can be made visible based on a user's physical location. This enables you to implement a location-based service. This is an optional value.</p> <p>To use this parameter, you implement the <code>requestBegin()</code> method in the <code>ParmHook</code> (request manager) interface. <code>requestBegin()</code> is called every time that the Wireless Edition receives a user request. You can place logic in this method that invokes an external position system. <code>requestBegin()</code> then passes the result in the request object. When the Wireless Edition invokes the service, it calls <code>isRuntimeVisible()</code>, passing it the user's position. You can test the position against the Area ID configured for the master service in <code>isRuntimeVisible()</code> to determine whether to make the service available.</p>
Cost	The cost-per-service access. If you set a value for this parameter, and configure the System.properties file to enable transaction logging, the Wireless Edition writes billing information to a transaction log. This is an optional value.
Description	An optional description of the master service.

After you have entered the desired values, click Next. The Adapter Selection screen appears.

Figure 3–6 The Selecting Adapter Screen

3.3.3 Step 3: Selecting an Adapter

The Adapter Selection screen displays the adapters available in the Wireless Edition repository. Click the desired adapter (for example, the SQLAdapter) and then click Next. The Initial Parameters screen appears.

3.3.4 Step 4: Setting the Init Parameters

The Init Parameters screen contains the init parameters for the adapter you chose in Step 3. Enter the values for these parameters in the adjacent fields and then click Next. The Input Parameters screen appears.

3.3.5 Step 5: Setting the Input Parameters

The Input Parameters Screen displays the input parameters for the adapter you selected in Step 3. The Master Service Creation Wizard queries the adapter definition to determine the parameters that appear in this screen. For this example, the Input Parameters screen displays the parameters for the Provisioning adapter. Every Wireless Edition parameter has the following attributes:

Table 3–3 Attributes of the Wireless Edition Adapters

Parameter	Value
Name	The name of the input parameter. The Wireless Edition Service Creation Wizard sets the name of the input parameter by querying the adapter definition.
Caption	The caption is the label that the Wireless Edition uses for the parameter when prompting for user input.
Comment	In the case of master services based on the Web Integration adapter, the Wireless Edition automatically populates this cell with the name of the WIDL service that uses the parameter. For services 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 at the Personalization Portal. 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 email addresses and personal identification numbers.
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. Other formats include: <ul style="list-style-type: none"> ■ A, for entry of uppercase letters or punctuation. ■ a, for entry of lowercase letters or punctuation. ■ N, for entry of numbers. ■ X, for entry of uppercase letters. ■ x, for entry of lowercase letters. For a complete list of formats, see the <i>Wireless Application Protocol Wireless Markup Language Specification, Version 1.1</i> .
Mandatory	Select this check box if this parameter must have a value. Remove the selection for optional parameters.

Parameter	Value
Value	<p>For most parameters, this value represents the default value for the parameter. If you specify a default value, the Wireless Edition does not prompt the user for a value. Default values can be overridden by a value specified by a service alias or, if the parameter is visible to the user, by the user at the Personalization Portal.</p> <p>The <code>PAsection</code> parameter is used by the Web Integration adapter. For <code>PAsection</code>, this value is the name of the WIDL service that the Web service should use. You can select the names from a drop-down selection list. If you do not specify a value for <code>PAsection</code>, the Wireless Edition service includes all WIDL services in the WIDL interface.</p>

From this screen, you can add or delete the input parameters for the adapter. To add an input parameter, highlight the parameter and click the Add button. To delete an input parameter, select the input parameter and click the Delete button. Clicking Reset sets the input parameters back to their original state. After you have finished adding or deleting the input parameters for the adapter, click Next. The Output Parameters screen appears.

3.3.6 Step 6: Setting the Output Parameters

The Output Parameters screen enables you to add the output parameters for the adapter. The output parameters displayed below are for the Provisioning adapter, the adapter selected in Step 3. The Master Service Creation Wizard queries the adapter definition to determine the parameters that appear in this screen. The output parameters have the following attributes:

Table 3–4 Output Parameters for the Wireless Edition Adapters

Parameter	Value
Name	The name of the output parameter. The Master Service Creation Wizard sets the name of the output parameter by querying the adapter definition.
Caption	The caption is the label that Wireless Edition uses for the parameter when prompting for user input.

Parameter	Value
Comment	<p>In the case of master services based on the Web Integration adapter, Wireless Edition automatically populates this cell with the name of the WIDL service that uses the parameter.</p> <p>For services based on other adapters, you can use this column to document the parameter. The comment is only used internally.</p>
User Customizable	<p>Specifies whether the end user can set a value for this parameter at the Personalization Portal. You can make most output 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 email addresses and personal identification numbers.</p>
Format	<p>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.)</p> <p>The default format is *M. Other formats include:</p> <ul style="list-style-type: none"> ■ A, for entry of uppercase letters or punctuation ■ a, for entry of lowercase letters or punctuation ■ N, for entry of numbers. ■ X, for entry of uppercase letters. ■ x, for entry of lowercase letters. <p>For a complete list of formats, see the <i>Wireless Application Protocol Wireless Markup Language Specification, Version 1.1</i>.</p>
Mandatory	<p>Select this check box if this parameter must have a value. Remove the selection for optional parameters.</p>
Value	<p>For most parameters, this value represents the default value for the parameter. If you specify a default value, the Wireless Edition does not prompt the user for a value. Default values can be overridden by a value specified by a service alias or, if the parameter is visible to the user, by the user at the Personalization Portal.</p> <p>The PAsession parameter is used by the Web Integration adapter. For PAsession, this value is the name of the WIDL service that the Web service should use. You can select the names from a drop-down selection list. If you do not specify a value for PAsession, the Wireless Edition service includes all WIDL services in the WIDL interface.</p>

To add an output parameter, highlight the output parameter and click the Add button. To delete an output parameter, select the output parameter and click the Delete button. Clicking Reset sets the output parameters back to their original state.

After you have finished adding or deleting the output parameters for the adapter, click Next. The Confirmation screen appears if the Wireless Edition has not found a `PASection` in the master service you have created. Review the values listed on the Confirmation screen. If they are correct, click the Finish button. You have completed creating a master service.

If your master service contains a `PASection`, the Create Result Transformer screen appears.

3.3.7 Step 7: Creating the Result Transformer (Optional)

After you have set the output parameters for the adapter, the Wireless Edition checks if the input parameters include `PASection`, the value used by the WIDL adapter to identify the service that is the entry point in the chained service sequence. If the Master Service Creation Wizard finds a `PASection` input adapter, it invokes the Create Result Transformer screen. The tabs in the Result Transformer panel represent different `PASections`.

To edit a `PASection`:

1. Click the tab that represents the `PASection` you wish to edit. Each panel contains a text editor for entering the XSLT style sheet. You can also import an XLST style sheet by clicking the import button.
2. Click Next after you have completed editing the XLST style sheet. The Device Transformer Screen appears. Leave this screen blank if you do not wish to create a result transformer and click Next until you reach the Confirmation screen.
3. If the values appear correct, click Finish to complete the creation of the master service.

3.3.8 Step 8: Setting the Device Transformer (Optional)

The Device Transformer Screen lists the logical devices in the repository. This screen enables you to specify the custom transformer used by the master service for a logical device. A custom transformer enables you to optimize the presentation of service content for a particular device. Since the transformer is specialized for a particular device and master service, you can associate a custom transformer with only one master service and one logical device.

To select a logical device:

1. Select the device for the master service you have created in Steps 1 through 6.
2. Click Next. The confirmation screen appears with a display of the parameters of the master service you have created.
3. Review the values listed on the Confirmation Screen. If they are correct, click Finish to complete the creation of a master service.

3.4 Creating a Location-Based Service

You use both the Service Designer and the Region Modeling Tool to create services that are visible to users only at specific locations.

In the Service Designer, you access the Region Modeling tool by selecting the Location Dependent check box in the either the General tab (used for modifying a folder or service in the Wireless Edition repository tree), or the *Create New Services* or the *Create New Folders* forms (invoked by right-clicking the Wireless Edition repository) and then by clicking the Browse button.

Note: Location independent services are services, such as email, which are always visible to the user at any location. These services are created by not selecting the Location Dependent check box.

In the Master Service Creation Wizard, you access the Region Modeling Tool by selecting the Browse button in the *Creating New Folder* and *Setting Service Properties* screens. For more information on using the Region Modeling Tool, see [Chapter 4, "Using the Region Modeling Tool"](#). For a further description of creating a location-based service, see [Section 9.6, "Creating Location-Based Services"](#) in [Chapter 9, "Walkthroughs"](#).

Note: You must be connected to the spatial database to return objects using the Region Modeling Tool.

3.4.1 Using the Service Designer to Create Location-Based Services

To make an existing folder or service location-based:

1. In the Wireless Edition repository tree, click the folder or service to which you wish to assign a region. The General tab appears.
2. In the General tab, click the Location Dependent check box. The Browse button appears.

3. Click the Browse Button. The Region Modeling Tool appears.
4. Right-click the region or regions that you wish to assign to the service or folder. Click Select.

Note: You can select multiple regions using <CTRL> + click.

5. The General tab reappears, showing the Area ID of the selected region.

To create a location-based folder or service from the Service Designer:

1. In the Wireless Edition repository tree, right-click the folder or service to which you wish to assign a region.
2. In the *Create New Folders* or *Create New Services* screens, click the Location Dependent check box. The Browse button appears.
3. Click the Browse Button. The Region Modeling Tool appears.
4. Right-click the region or regions that you wish to assign to the service or folder. Click Select.

Note: You can select multiple regions using <CTRL> + click.

5. The *Create New Folders* or *Create New Services* screen reappears, showing the Area ID of the selected region.

3.4.2 Using the Master Service Creation Wizard to Create Location-Based Services

To access the Region Modeling Tool from either the Create the Master Service Creation Wizard:

1. In either the *Creating New Folder* and *Setting Service Properties* screens, click the Browse button. The Region Modeling Tool appears.
2. In the left frame, select the region node to which you wish to assign a service. For example, from the SDR hierarchy, expand the North America node to the California node, and then expand to the San Francisco node.
3. Right click the region or regions (for example, San Francisco).

Note: You can select multiple regions using <CTRL> + click.

4. Click Select. The Creating Folder screen reappears, showing the Area ID of the selected node (for example, 8686, the Area ID of San Francisco). A location has been assigned to the service folder.

3.5 Modifying a Master Service

To modify a master service using the Service Designer, click the master service that you want to change. The object's property tabs appear in the right-hand panel of the Service Designer. You can enter new values directly in the parameter fields, then click Apply to save your changes.

Master services have the following property panels:

- General
- Init Parameters
- Input Parameters
- Output Parameters
- Result Transformer
- Device Transformer

3.5.1 General Panel

The General panel contains parameters that identify and describe the master service. It includes fields for the service name, the adapter on which the service is based, and the cost of the service. Typically, you set the properties listed in this panel when you first created the master service.

3.5.2 Init Parameters Panel

The Init Parameters panel shows the initialization parameters for the adapter. These parameters vary depending on the adapter implementation. The Service Designer generates the fields in this panel from the adapter definition for the master service. When the Wireless Edition first invokes the adapter, it passes the values you set in this panel to the adapter.

The following sections describe the initialization panels for master services based on the adapters provided by the Wireless Edition.

3.5.2.1 Web Integration Init Parameters

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 Wireless Edition 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.

For a master service based on the Web Integration adapter, the Init Parameters panel appears as follows:

The Init Parameters panel contains the following parameters:

Table 3–5 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 Wireless Edition server reside on the same machine, use <code>localhost:port</code> . This field is required. The server you specify in this field must be running for the Service Designer 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.

3.5.2.2 SQL Adapter Init Parameters

Given a URL, the stripper adapter dynamically retrieves and converts the content of the URL target. Unlike the Web Integration adapter, which uses a predefined mapping of the source content, the stripper adapter dynamically processes the markup tags in the content. Currently, the stripper adapter either removes the original markup tags or leaves them intact. Developers can extend the stripper adapter, however, so that it processes the tags in another way.

For a master service based on the SQL adapter, the Init Parameters panel includes the following parameters:

Table 3–6 Init Parameters for the SQL Adapter

Parameter	Value
The Statement	<p>The actual SQL statement that invokes the query, PL/SQL procedure, or stored procedure.</p> <p>Note: The SQL statement should be entered without a semicolon.</p> <p>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:</p> <pre>begin mypackage.foo(:expr); end;</pre> <p>Where <code>:expr</code> is the name of the variable. You must define the parameter manually in the input panel.</p>
Type of Statement	<p>The type of SQL statement used by the master service. Allowable values:</p> <p>QUERY: for a select statement. This type of statement returns a Simple Result document. You can use output filtering with QUERY statements. For information on filtering output, see the <i>Oracle9i Application Server Wireless Edition Developer's Guide</i>.</p> <p>PLSQL: to use a PL/SQL procedure. This type of statement returns results to a database buffer.</p> <p>CALL: to run a stored procedure (SQL92 syntax only). This returns either a Simple Result or an Adapter Result element.</p>
Password	The password of the database user.
Username	The name of the database user.
JDBC Driver	The type of JDBC driver used to access the SQL data source.
JDBC Connect String	The database connect string for the database to access. For example, to access an Oracle database using the thin driver, use the connect string: <code>jdbc:oracle:thin:@domain:port:SID</code>

3.5.2.3 URL Adapter Init Parameters

The URL adapter enables integration with existing Java, JSP, or XML applications. The URL adapter supports parameterization and cookie management, thus providing an easy way to call Wireless Edition services from remote applications and make them wirelessly available. The URL adapter is well suited for developers

wishing to build a wireless Web site or internet application using only Java servlets and Java Server Pages (JSPs).

Note: The target attributes in the SimpleResult XML documents retrieved by the URL adapter must be URL-encoded.

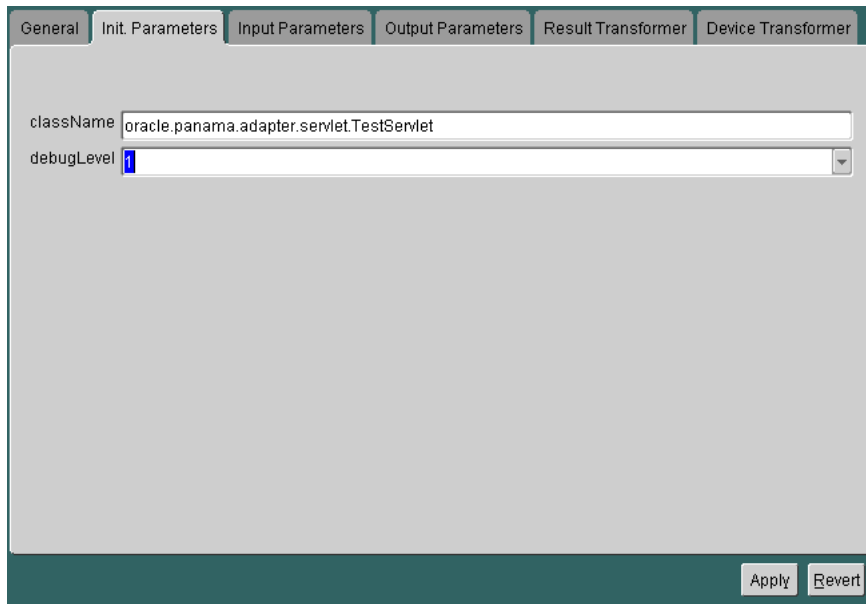
There are no Init parameters for this adapter.

3.5.2.4 Servlet Adapter Init Parameters

The Servlet adapter enables developers to integrate other applications that are already Java servlets, providing a convenient way to call them as Wireless Edition services and make them wirelessly available.

For a master service based on the servlet adapter, the Init Parameters panel appears as follows:

Figure 3–7 *The Init Parameters Panel for the Servlet Adapter*



The panel contains the following parameters:

Table 3–7 Init Parameters for the Servlet Adapter

Parameter	Value
className	The complete class name of the Java servlet file. For example: oracle.panama.adapter.servlet.TestServlet.
debugLevel	The debug level. Possible values are: 0, for none; 1, for notifications; and 2 for trace.

3.5.2.5 Mail Adapter Init Parameters

The Mail adapter is based on the JavaSoft's Java Mail API and provides mail services to both POP3 and IMAP4 mail servers. It provides basic authentication and mail session management. The Mail adapter enables users to browse, create, delete, reply, and forward mail and folders from their wireless devices. For more information on the Mail adapter, see [Appendix A, "The Mail Adapter"](#).

The Mail adapter has no init parameters. See [Section 3.5.3.2](#) for the input parameters of the Mail adapter.

3.5.2.6 Directory Adapter Init Parameters

The Directory adapter provides directory service and is based on the JavaSoft Java Naming Directory Interface (JNDI) 1.2 specification. The Directory adapter also enables you to create hyperlinks.

The Init Parameters panel for the Directory Adapter includes the following parameters:

Table 3–8 Init Parameters for the Directory Adapter

Parameter	Value
Host Name	The name or IP address of the machine on which the Directory server runs. All directory queries are directed to the directory server running on this machine. If no host name is specified, the address defaults as localhost.
Port number	This defaults to port 389.
User Name	Name of the database user.
Password	Password of the database user.
Query Strings	The LDAP query strings that a provider of the service should specify in order to provide the end user with a menu of the query list.

Parameter	Value
HotLink String	This is an optional entry. The HotLink strings that may be used to qualify an output attribute so that the attribute will be displayed as a hyperlink in the end user's browser screen.
ResultType	The value for ResultType is SimpleResult.

3.5.3 Input Parameters Panel

The Input Parameters panel displays the input parameters for the adapter. The Service Designer queries the adapter definition to determine the parameters that appear in this panel. The master service 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 service in the WIDL interface (`PAsection`), are set by the master service or master service alias. `PAsection` is an internal parameter, not exposed to the end user. In addition to `PAsection`, the Wireless Edition provides these input parameters:

Table 3–9 *Input Parameters*

Variable	Value
<code>PAservicepath</code>	The relative path to a Wireless Edition service, <code>/UsersFolders/joe/myChain</code> , for example.
<code>PAdebug</code>	The debugging option. If true (set to 1), the Wireless Edition produces verbose output to the log files. In this case, in addition to notifications and warnings, the Wireless Edition writes the results of adapter invocations to the log file. This enables you to examine service content in its internal, XML format, which can help you to create result transformers and solve service and transformer problems.
<code>PAsection</code>	The WIDL adapter uses this value to identify the service that serves as the entry point in the chained service sequence.
<code>PAuserid</code>	The user name.
<code>PApassword</code>	The user password.
<code>PAsid</code>	The Wireless Edition session identifier.

You can configure your parameters in the Service Designer. Every Wireless Edition parameter has the following attributes:

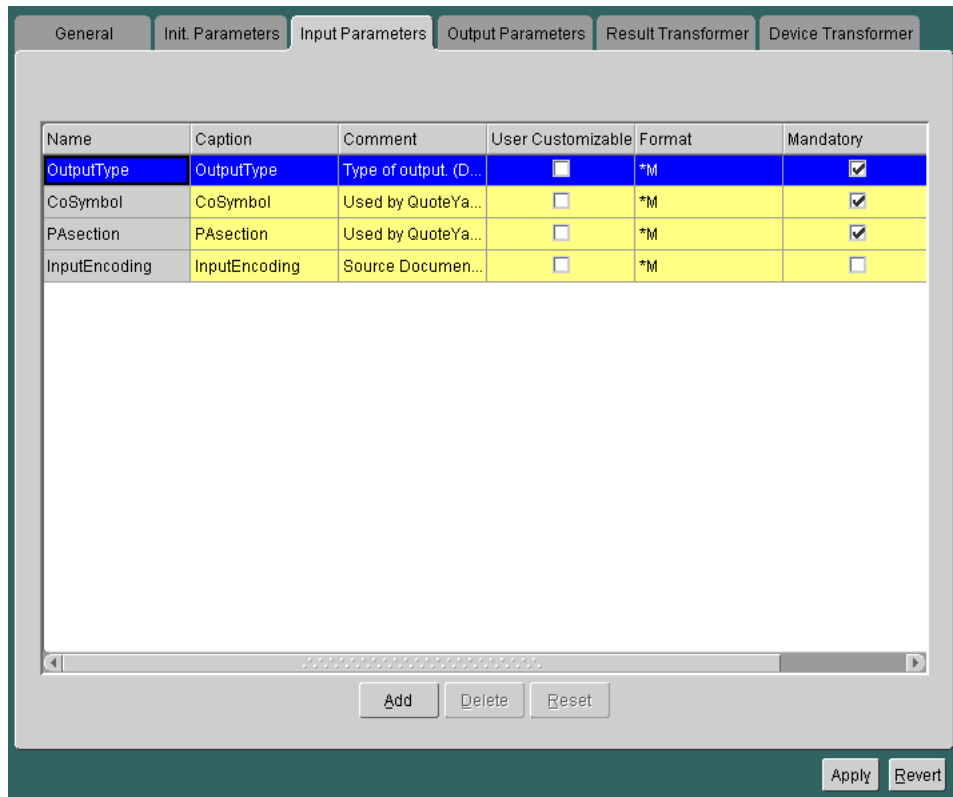
Table 3–10 Input Parameters Attributes

Parameter	Value
Name	The name of the input parameter. The Service Designer sets the name of the input parameter by querying the adapter definition.
Caption	The caption is the label that the Wireless Edition uses for the parameter when prompting for user input.
Comment	In the case of master services based on the Web Integration adapter, the Wireless Edition automatically populates this cell with the name of the WIDL service that uses the parameter. For services 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 at the Personalization Portal. 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 email addresses and personal identification numbers.
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. Other formats include: <ul style="list-style-type: none"> ■ A, for entry of uppercase letters or punctuation ■ a, for entry of lowercase letters or punctuation ■ N, for entry of numbers. ■ X, for entry of uppercase letters. ■ x, for entry of lowercase letters. For a complete list of formats, see the <i>Wireless Application Protocol Wireless Markup Language Specification, Version 1.1</i> .
Mandatory	Select this check box if this parameter must have a value. Remove the selection for optional parameters.

Parameter	Value
Value	<p>For most parameters, this value represents the default value for the parameter. If you specify a default value, the Wireless Edition does not prompt the user for a value. Default values can be overridden by a value specified by a service alias or, if the parameter is visible to the user, by the user at the Personalization Portal.</p> <p>The <code>PAsection</code> parameter is used by the Web Integration adapter. For <code>PAsection</code>, this value is the name of the WIDL service that the Web service should use. You can select the names from a drop-down selection list. If you do not specify a value for <code>PAsection</code>, the Wireless Edition service includes all WIDL services in the WIDL interface.</p>

3.5.3.1 Web Integration Input Parameters

The following figure shows a sample Inputs panel for a Web service.

Figure 3–8 The Input Parameters Panel for the Web integration Adapter

The master service determines the parameters to 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 services within the WIDL interface.

In addition to the custom input parameters that you create, Web Integration services provide these parameters:

- OutputType
- PAsession
- 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. If returning 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 service. You use `RawResult` for chained services.

`PAsection` is the name of the WIDL service that you want the master service to invoke. A WIDL interface can include more than one WIDL service. The Wireless Edition lists the WIDL service 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 service. See [Chapter 9, "Walkthroughs"](#) for information on creating a Web service. 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. It is part of the multi-byte character support. The *Oracle9i Application Server Wireless Edition Developer's Guide* provides more information about multi-byte character support.

3.5.3.2 Mail Adapter Input Parameters

The Mail adapter has the following input parameters.

Table 3–11 *Init Parameters for the Mail Adapter*

Parameter	Value
<code>mail_type</code>	The mail server type. Options are IMAP and POP
<code>Incoming_Server</code>	The host name for the incoming server (for example, <code>gmmail.oracle.com</code>).
<code>Outgoing_Server</code>	The host name of the outgoing mail server (for example, <code>gmsmtp01.oraclecorp.com</code>)
<code>mail_ReturnDomain</code>	The domain name of the user's email address. This domain must start with the @ symbol (for example, <code>@oracle.com</code>).
<code>UserName</code>	The user name of the email account.
<code>Password</code>	The password used to login to the email account.

3.5.3.3 SQL Master Service Input Parameters

You can configure SQL input parameters just as you can Web service parameters. You specify input parameters in the SQL statement you use to implement the service. For information on creating parameterized SQL services, see [Section 3.5.2.2](#).

3.5.3.4 Stripper Adapter Input Parameters

For a master service based on the stripper adapter, the Inputs panel appears as follows:

Figure 3–9 *The Input Parameters Panel for the Stripper Adapter*

Name	Caption	Comment	User Customizable	Format	Mandatory	Value
url	url	Target URL	<input type="checkbox"/>	*M	<input checked="" type="checkbox"/>	
beginTag	beginTag	Match begin string	<input type="checkbox"/>	*M	<input checked="" type="checkbox"/>	
endTag	endTag	Match end string	<input type="checkbox"/>	*M	<input checked="" type="checkbox"/>	
stripLevel	stripLevel	Strip filter#	<input type="checkbox"/>	*M	<input checked="" type="checkbox"/>	
title	title	Result title	<input type="checkbox"/>	*M	<input checked="" type="checkbox"/>	

The panel contains the following parameters:

Table 3–12 *Input Parameters for the Stripper Adapter*

Parameter	Value
url	The URL of the source page. If you do not include the protocol in the URL, the Wireless Edition adds "http://" before this value with
beginTag	A string that matches the start point of the text retrieved from the URL target page. This value is optional. If not specified, the start point is the beginning of the page.

Parameter	Value
endTag	A string that matches the end point of the text retrieved from the URL target page. This value is optional. If not specified, the end point is the end of the page.
stripLevel	<p>The strip level identifies the class used by the stripper adapter to filter the retrieved content. The Wireless Edition provides two levels:</p> <ul style="list-style-type: none"> ■ 0: Retains all markup tags in the content. ■ 1: Removes all markup tags in the content. <p>You can implement additional strip levels or filters. For more information, see the <i>Oracle9i Application Server Wireless Edition Developer's Guide</i>.</p>
title	An optional title of the result.

3.5.3.5 URL Adapter Input Parameters

The URL adapter allows applications to invoke other URLs and pass in arguments. In addition, the URL adapter enables the following functions:

- Integration with existing Java, JSP, or XML applications.
- Parameterization and cookie management, enabling remote applications to become wirelessly available.
- Support for the GET/POST method.

The URL adapter is well suited for developers wishing to build a wireless Web site or internet application using only Java servlets and Java Server Pages (JSPs).

The panel contains the following parameter:

Table 3–13 Input Parameters for the URL Adapter

Parameter	Value
URL	<p>The URL of the XML document that is retrieved by the URL Adapter. The XML document should conform to the SimpleResult DTD.</p> <p>For example, if the URL parameter is <code>file:///d:/SimpleResultExample1.xml</code>, then the adapter gets the file <code>d:/SimpleResultExample1.xml</code> from the hard drive on the local machine. If the URL parameter is <code>http://ptg1.oracle.com/SimpleResultExample2.xml</code>, then the adapter gets the file <code>SimpleResultExample2.xml</code> from the host <code>ptg1.oracle.com</code>.</p>

3.5.3.6 Servlet Adapter Input Parameters

For a service that is based on the Servlet adapter, the Input Parameters panel appears as follows:

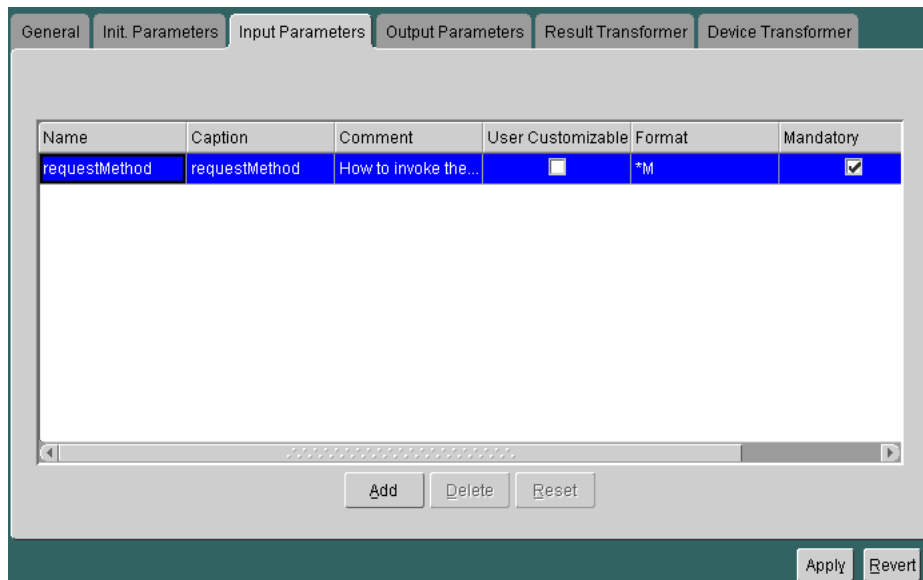
Figure 3–10 The Input Parameters Panel for the Servlet Adapter

Table 3–14 Input Parameters for the Servlet Adapter

Parameter	Value
requestMethod	<p>The value of this parameter can be any of the following: GET, HEAD, POST, PUT, DELETE, OPTIONS, or TRACE.</p> <p>Depending on the value of this parameter, the appropriate method of the servlet gets called when the adapter is invoked.</p> <p>For example, if the value of requestMethod is GET, then the doGet method of the servlet is called.</p>

3.5.4 Output Parameters Panel

The output parameters panel allows you to set captions for service output parameters.

3.5.5 Result Transformer Panel

The Result Transformer panel specifies a transformer that the Wireless Edition uses to convert Adapter Result content. The Wireless Edition provides two content formats, Adapter Result and Simple Result. Adapter Result is intended to be an intermediary format for passing raw data between services. Device transformers, which convert service content for the target format, cannot convert Adapter Result format. A result transformer must therefore convert the content to Simple Format before it can be processed by a device transformer.

To create a result transformer:

1. Click Add.
2. Enter the name of the result transformer in the name field. You must use the same name for the result transformer as you used for the WIDL service to which it applies. This is the value of the `PAsection` input parameter.
3. Click Edit. The XSL Editor window appears. The XSL Editor is a simple text editor you can use to build your result transformer. You can cut and paste from other environments into the XSL Editor.
4. When finished, click OK.

3.5.6 Device Transformer Panel

The Device Transformer panel lists the logical devices in the repository. You can specify a custom transformer to be used with the master service for a logical device. A custom transformer enables you to optimize the presentation of service content

for a particular device. Since the transformer is specialized for a particular device and master service, you can associate a custom transformer with only one master service and one logical device.

3.6 Deleting a Master Service

You can delete a master service in the Service Designer as follows:

1. Select the master service that you want to delete.
2. Right-click and select Delete.
3. Confirm the action.

When you delete a master service, the Service Designer flags any aliases to the master service.

3.7 Creating a Folder

You can create folders in either the Master Services folder or the Service Trees folder of the Wireless Edition repository tree. To create a folder, you use the Create New Folder form.

To invoke the form:

1. Select the root folder under which you want to place the folder.
2. Right click.
3. Select Create New Folder.

The Create New Folder form appears as follows:

Figure 3–11 The Create New Folder Form

The panel includes the following parameters:

Table 3–15 Parameters for the Create New Folder Form

Parameter	Value
Name	The folder name must be a unique name in the parent folder.
Valid	Select the Valid check box to enable the folder.
Visible	Select the Visible check box to make the folder accessible to users.
Owner	The user or group that owns the folder. Only the owner can modify the folder.
Location Dependent	Selecting this check box invokes the Browse button, which gives you access to the Region Modeling Tool.
Browse	Clicking this button gives you access to the Region Modeling Tool. You use the Region Modeling Tool to assign a region to the service.

Parameter	Value
Sequence	<p>This integer value lets you alter the order in which services 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 services and folder appear.</p> <p>By default, the Wireless Edition sorts services and folders in ascending order by sequence number, then by name. You can change this behavior by altering the <code>order.services</code> property in the System.properties file. You can specify sorting by sequence number, name, or date last written, in either ascending or descending order.</p>
Area Id	An integer value that allows you to specify whether a folder is visible based on a user's physical location. This enables you to implement location-specific services. This is an optional value.
Description	An optional description of the folder.

3.8 Deleting a Folder

You can delete a folder in the Service Designer as follows:

1. Select the folder that you want to delete.
2. Right-click and select Delete.
3. Confirm the action.

3.8.1 Service Trees

A service tree is a folder you use to organize and distribute access to services. It can contain master services, service aliases, or other folders.

Folders make services accessible to users. When you create a new user, you also create a service tree for that user. Any service that you place in a user's service tree (or user's home), is accessible to the user.

Similarly, when you create a group, you specify the service trees that belong to the group. Any member of the group can access the services in that group's service trees. While a user can have only one private service tree, groups can have many service trees. Different groups can share the same service tree.

If you do not specify an owner when you create a service tree, the tree is public. If you specify an owner, the service tree is private. End users can access any public

service tree, but they can only access their own private service trees. They have complete control of their service trees from the Personalization Portal; they can copy services between service trees, rename services, and create folders.

The Wireless Edition does not provide dependency tracking for a private service (a service in a private service tree). If the target service of the private service is no longer valid, an error occurs when a user attempts to submit a request for that service. The Wireless Edition then advises the user to delete the service from the private service tree.

3.9 Creating a Service Alias

A service alias is a link to a master service, folder, or other alias. Service aliases help you to distribute service access to multiple users or groups. They also enable you to specialize master services, since default parameter values specified by an alias override values set at the master service. This characteristic provides several benefits. One benefit is that it enables you to localize services. For example, suppose you create a service that delivers restaurant information for a city. The adapter takes a single parameter, a location, and returns a list of restaurants in the area. While the master service can specify a more general location, such as the city, you can create aliases that provide a more specific parameter, such as a district within the city. You can then distribute the aliases, as appropriate, to user groups that you assemble based on the users' locations of residence.

To create a service alias:

1. Highlight and right-click any Service Tree subfolder in the Wireless Edition Repository tree.
2. Click Create New Alias.

The first form in the sequence appears as follows:

Figure 3–12 Create New Alias Form (First Form)

The panel includes the following parameters:

Table 3–16 Parameters for the Create New Alias Form

Parameter	Value
Name	The service alias name. This must be a unique name within the parent folder.
Service	The master service referenced by the service alias. To select a master service, click Browse and choose a master service from the Browse Services window.
Valid	Select the Valid check box to enable the alias.
Visible	Select the Visible check box to make the alias accessible to users.
Owner	The user or group that owns the alias. Only the owner can modify the alias. If you are creating an alias in a user's home folder, you can make the owner that user. This enables the user to modify the service—by renaming it, for example—at the personalization portal.

Parameter	Value
Sequence	<p>This integer value lets you alter the order in which services 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 services and folders appear.</p> <p>By default, the Wireless Edition sorts services and folders in ascending order by sequence number, then by name. You can change this behavior by altering the <code>order.services</code> property in the System.properties file. You can specify sorting by sequence number, name, or date last written, in either ascending or descending order.</p>
Description	An optional description of the alias.

Complete the properties in the form and click Finish to create the alias in the repository. You do not need to complete the second form in the sequence. You can configure the runtime parameters for the alias by modifying the alias properties, just as you would for the master service on which the alias is based. For more information on specifying runtime parameters, see [Section 3.5.3](#).

Using the Region Modeling Tool

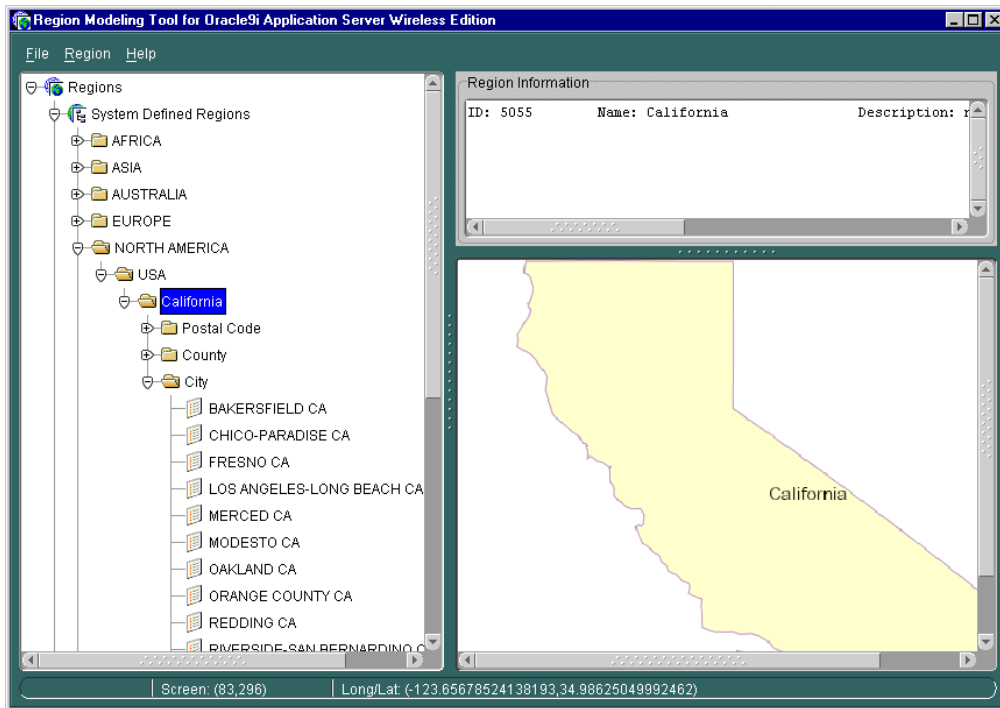
This document describes the Region Modeling Tool. Each section of this document presents a different topic. These sections include:

- [Section 4.1, "Overview"](#)
- [Section 4.2, "Accessing the Region Modeling Tool"](#)
- [Section 4.3, "Managing Spatial Objects"](#)
- [Section 4.4, "Creating Location-Based Master Services and Folders"](#)
- [Section 4.5, "Creating Location-Based Services with Location-Specific Content"](#)

4.1 Overview

The Wireless Edition enables you to create location-based services using the Region Modeling Tool. The Region Modeling Tool enables you to create services that are visible to Wireless Edition users at specific locations.

The Region Modeling Tool enables developers to create these location-based services by assigning a location to a folder or to a master service from the spatial data repository. The tool enables you to view a spacial object as a map image with its corresponding geometry. In addition, the tool enables you to access, view, and modify the data stored in the spatial database in the Wireless Edition repository.

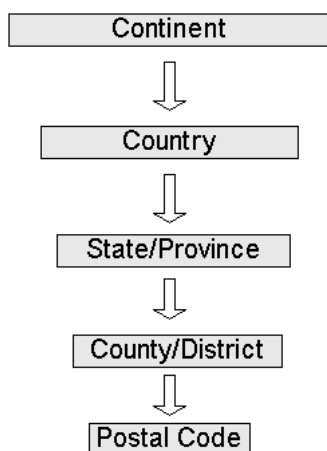
Figure 4–1 The Region Modeling Tool

The Region Modeling Tool presents spatial data as a set of regions, which are divided into system-defined regions (SDRs) and user-defined regions (UDRs).

4.1.1 The System-Defined Region Hierarchy

The SDRs are administrative boundaries organized in a hierarchy of subregions as follows:

Figure 4–2 Hierarchy of the System-Defined Region



The left frame of the Region Modeling tool presents the SDRs as a hierarchical directory organized by continent, country, state, county, and postal code nodes. Clicking one of these nodes displays its corresponding map and geometry (longitude/latitude) in the right frame, and its identifier (ID: 5055 in Figure 3-6, above), name, and description in the top frame.

In a typical SDR hierarchy, a continent node contains country nodes. Country nodes contain state nodes, and state nodes contain separate child folders for city, postal code, and county. If a country lacks states or provinces, then the country node has separate child folders for city, postal code, and county. If state data exists in any country, then all data for the postal code, city, or county must be contained in a state node.

Note: You cannot have state data and data for postal code, city, or county under country node.

Except for the continent region, which is at the top of the hierarchy, every item in the table of the hierarchy carries the region identifiers of its parent regions. For example, a county region carries the state identifier as well as the country identifier.

The Region Model Data Schema

The following is a description of a region model data schema.

Table 4–1 The Region Model Data Schema

Object	Value
Continent	
ID	NOT NULL NUMBER
NAME	VARCHAR2(100)
DESCRIPTION	VARCHAR2(2000)
GEOMETRY	MDSYS.SDO_GEOMETRY
REFCNT	NUMBER
Country	
ID	NOT NULL NUMBER
NAME	VARCHAR2(300)
CONT_ID	NUMBER
DESCRIPTION	VARCHAR2(2000)
GEOMETRY	MDSYS.SDO_GEOMETRY
REFCNT	NUMBER
State	
ID	NOT NULL NUMBER
NAME	VARCHAR2(400)
CONT_ID	NUMBER
COUNTRY_ID	NUMBER
DESCRIPTION	VARCHAR2(2000)
GEOMETRY	MDSYS.SDO_GEOMETRY
REFCNT	NUMBER
County	
ID	NOT NULL NUMBER
NAME	VARCHAR2(400)
COUNT_ID	NUMBER
COUNTRY_ID	NUMBER
STATE_ID	NUMBER

Object	Value
DESCRIPTION	VARCHAR2(2000)
GEOMETRY	MDSYS.SDO_GEOMETRY
REFCNT	NUMBER
City	
ID	NOT NULL NUMBER
NAME	VARCHAR2(400)
COUNT_ID	NUMBER
COUNTRY_ID	NUMBER
STATE_ID	NUMBER
DESCRIPTION	VARCHAR2(2000)
GEOMETRY	MDSYS.SDO_GEOMETRY
REFCNT	NUMBER
Postalcode	
ID	NOT NULL NUMBER
NAME	VARCHAR2(400)
COUNT_ID	NUMBER
COUNTRY_ID	NUMBER
STATE_ID	NUMBER
DESCRIPTION	VARCHAR2(2000)
Userdefined	
ID	NOT NULL NUMBER
NAME	VARCHAR2(400)
REFCNT	NUMBER
TYPE	NUMBER
PARENT_FOLDER_ID	NUMBER
DESCRIPTION	VARCHAR2(2000)
GEOMETRY	MDSYS.SDO_GEOMETRY

Note: When you log into the Wireless Edition repository, you also log into the spatial database.

4.1.2 User-Defined Regions

In addition to the SDRs, you can create your own regions, or user-defined regions (UDRs). These UDRs can be constructed in two ways:

- A point location based on an address or a region around an address.
- An aggregate of existing SDRs.

For example, the Region Modeling Tool allows you to create a service that is available ten miles around a specific address. You can make an aggregate of areas, such as the New England states of Maine, New Hampshire, Vermont, Massachusetts, and Rhode Island.

Unlike the SDRs, the UDRs are not arranged into a hierarchy; however, users can organize their UDRs by creating folders.

4.2 Accessing the Region Modeling Tool

There are two methods for accessing the Region Modeling Tool: you can access it in its Standalone mode through the Tools menu in the Service Designer, or in its browse mode when using the Service Designer to create a location-based master service or folder.

In the Standalone mode, you can create both UDRs and load data into SDRs. For more information on using the Region Modeling Tool in its Standalone mode, see [Section 4.3, "Managing Spatial Objects"](#).

When you access the Region Modeling Tool while creating a master service or folder, you access the tool in the browse mode. In the browse mode, you create location based services by assigning regions to services and folders. In addition, you can create UDRs and search for regions. For more information on using the Region Modeling Tool in its browse mode, see [Section 4.4](#).

4.3 Managing Spatial Objects

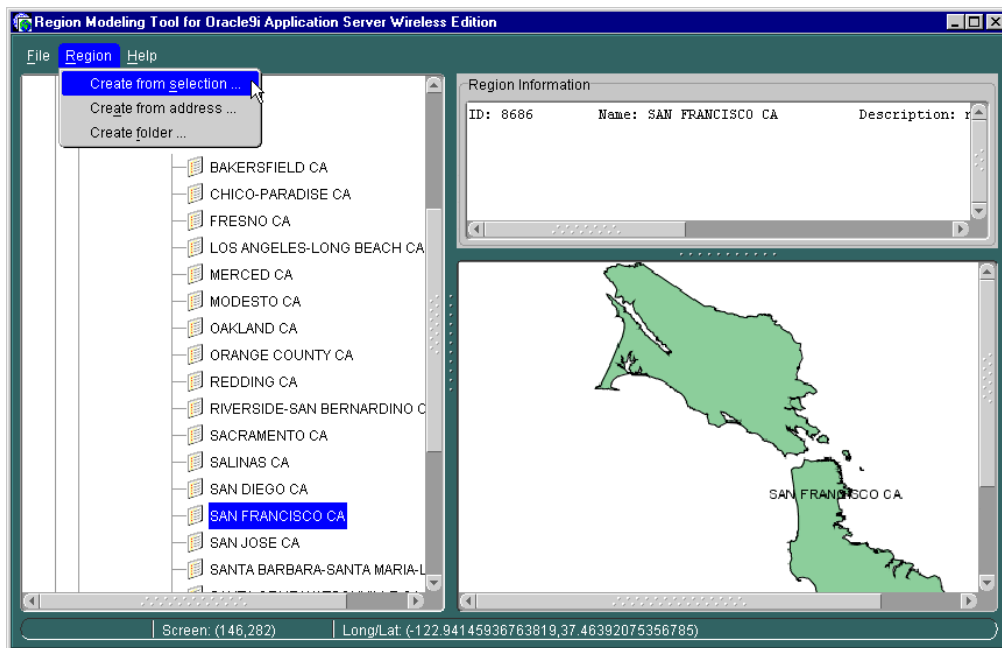
You use the Region Modeling Tool in its Standalone mode to view and manage regions.

4.3.1 Accessing the Region Modeling Tool in the Standalone Mode

To access the Region Modeling Tool in its Standalone mode:

1. Select the Tools Menu from the Service Designer.
2. Select Region Modeling Tool. The Region Modeling Tool appears.

Figure 4–3 The Region Modeling Tool with Region Menu Displayed



4.3.2 Creating User-Defined Regions

You can create a UDR by selecting from the SDR hierarchy, or you can create a UDR from an address.

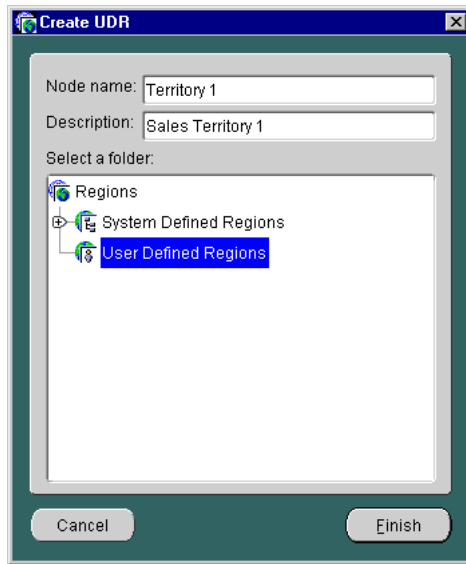
4.3.2.1 Creating a User-Defined Region from the SDR Hierarchy

To create a UDR using the SDR hierarchy:

1. Select the desired region or regions from the SDR hierarchy in the left frame. The Create UDR form appears.
2. Select the Create From Selection from the Region Menu.

Note: You can select multiple regions by using <CONTROL> + click.

Figure 4–4 The Create UDR Form

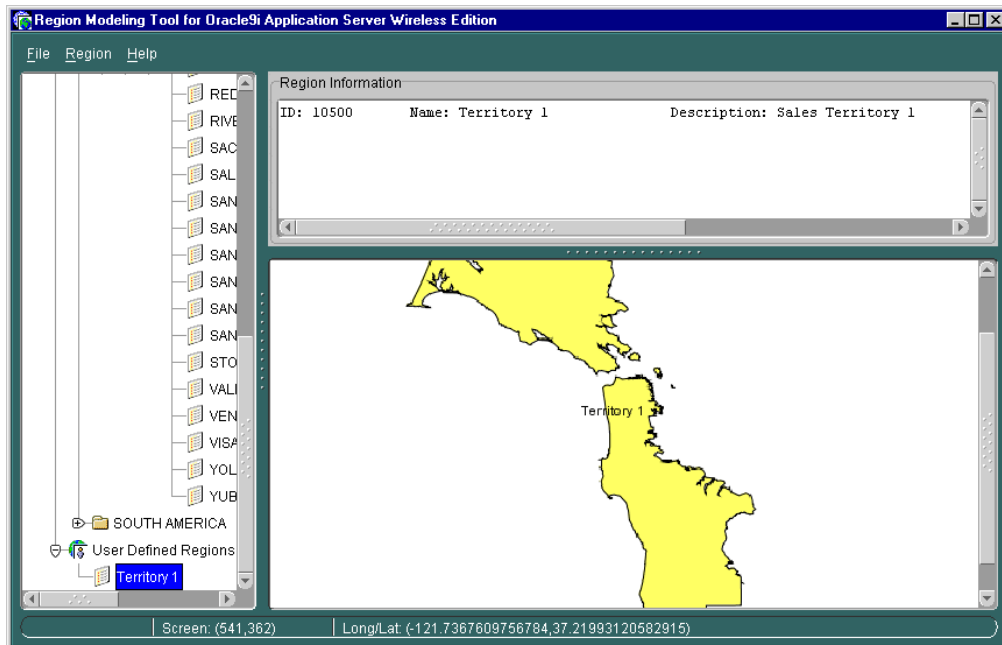


3. In the UDR form, enter a name for your region that is meaningful to you in the Node Name field. Likewise, add a description that is meaningful to you in the Description field. Click Finish.

Note: You can use the Create Folder function from the Region menu to create folders for your UDRs.

4. In the UDR hierarchy, click the UDR that you created. The map of your UDR displays in the right frame, and its ID, name, and description display at the top frame.

Figure 4–5 *The User-Defined Region*

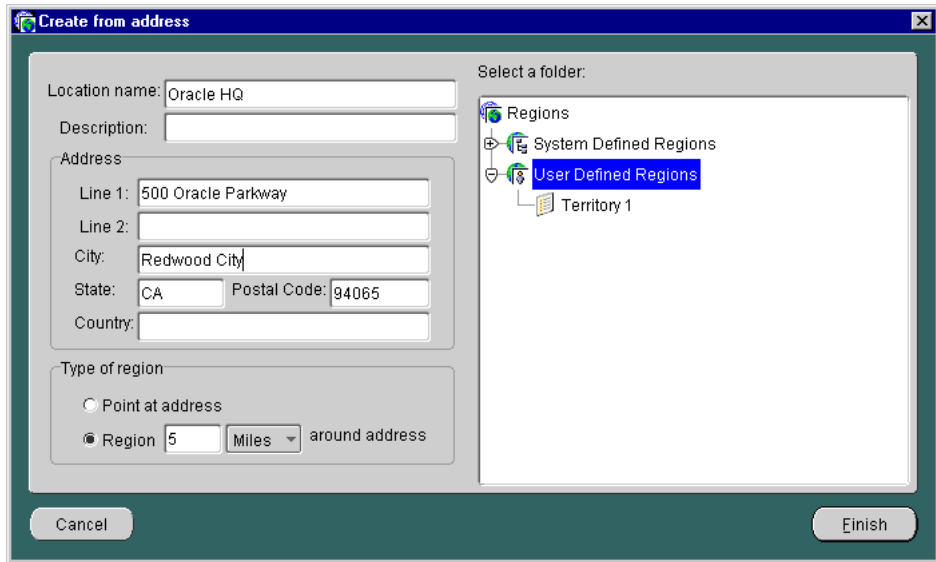


4.3.2.2 Creating a UDR from an Address

To create a UDR from an address:

1. From the Region menu, select Create From Address. The Create from Address screen appears.

Figure 4–6 The Create From Address Screen



2. Enter the name of the location in the Name field.
3. In the Address section of the screen, enter the exact address.
4. In the Type of Region section, select either Point at Address radio box for the exact location, or the Region radio box for a defined parameter around the address.
5. Click Finish to complete the creation of a UDR from an address.

Note: You must have your geocoding service running when creating UDRs from addresses.

4.3.3 Deleting a User-Defined Region

To delete a UDR:

1. Right-click the UDR in the left frame of the Region Modeling Tool.

2. Select Delete.

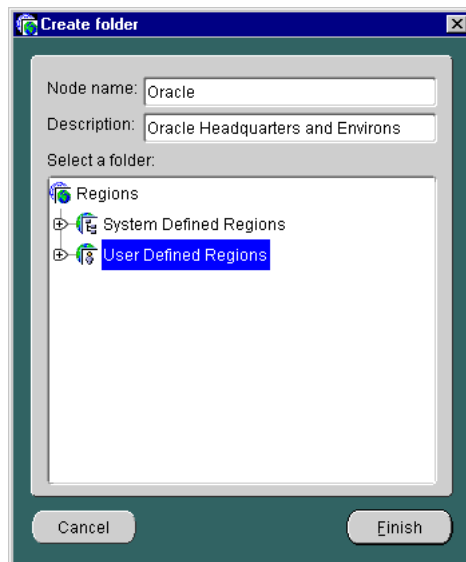
Note: You can only delete a region that has no services assigned to it. Regions with reference counts of 0 have no services assigned to them.

4.3.4 Creating a Folder

The UDRs are not automatically arranged in a hierarchy. They can be made easily accessible through the creation of folders. To create a folder:

1. Select the Create Folder option from the Region menu. The Create Folder appears.

Figure 4–7 The Create Folder Screen



2. Enter a name that is meaningful to you in the Node Name field.
3. Enter a description that is meaningful to you in the Description field.
4. Click Finish.

4.3.5 Deleting a Folder

To delete a folder:

1. Right-click a folder in the left frame of the Region Modeling Tool.
2. Select Delete.

Note: You can only delete a folder that contains UDRs with reference counts of 0, meaning that no services have been assigned to them. Likewise, if you select multiple folders for deletion, all of the folders selected must contain UDRs with reference counts of 0.

4.3.6 Creating System-Defined Regions

You create system-defined regions (SDRs) by loading spatial data into the Wireless Edition repository.

Note: The Wireless Edition repository contains a small set of built-in sample system-defined regions.

The following sections describe two methods of loading data into the repository:

- [Section 4.3.6.1, "Loading SDR Data Using SQL*Plus"](#)
- [Section 4.3.6.2, "Loading SDRs from the Region Modeling Tool"](#)

4.3.6.1 Loading SDR Data Using SQL*Plus

The following example of loading additional spatial data using SQL*Plus assumes that your data exists in a table and that the geometry is in the Oracle spatial format (MDSYS.SDO_GEOMETRY). An example of a table called `mystate` of US data is as follows:

Table 4–2 Example of `mystate` Table

Name	Geometry	Description
Alabama	{.....}
Alaska	{.....}
California	{.....}

Perform the following steps to load SDR data.

1. Use the following SQL statement to find out the country ID and continent ID of the US:

```
SELECT id, name, cont_id FROM Country;
...
5009 USA 5001
...
```

In this example, the country ID for USA is 5009, and the continent ID is 5001.

2. Use the following SQL statement to insert data from the table `mystate`:

```
INSERT INTO state
(ID, CONT_ID, COUNTRY_ID, NAME, DESCRIPTION, GEOMETRY, REFCNT)
SELECT idseq.nextval, 5001, 5009, name, description, geometry, 0 from mystate;
```

Note: The sequence name of the region ID is `idseq`. Also, if you are loading data for postal codes, cities, or counties directly under a country with a non-default hierarchy, you should use 0 as the `state_id` in the table for postal code, city, and county.

Formatting Geometry Data

Geometry data is assumed to be in WGS-84 format. If the geometry data is not in WGS-84 format, you need to transform it before loading using `MDSYS.SDO_CS.TRANSFORM`. The SRID for WGS-84 is 8307. For more information, refer to Oracle Spatial User's Guide and Reference.

Geometry data must be in Oracle 8.1.6 GTYPE or later. Otherwise, the user must migrate the data using `SDO_MIGRATE.FROM_815_TO81X`. For more information on GTYPE and the migration utility, refer to Oracle Spatial User's Guide and Reference.

4.3.6.2 Loading SDRs from the Region Modeling Tool

To load SRD data from the Region Modeling Tool:

1. Select the Region Modeling Tool from the Tools menu of the Service Designer.
2. Right-click a SDR, a menu appropriate to the the type and the status of the region appears. For example, right-clicking a continent node invokes the option to add country data to that continent hierarchy.

3. Add data by entering the appropriate information into the add data forms. The add data forms contain the following fields.

Table 4–3 Fields in the Add Data Screens

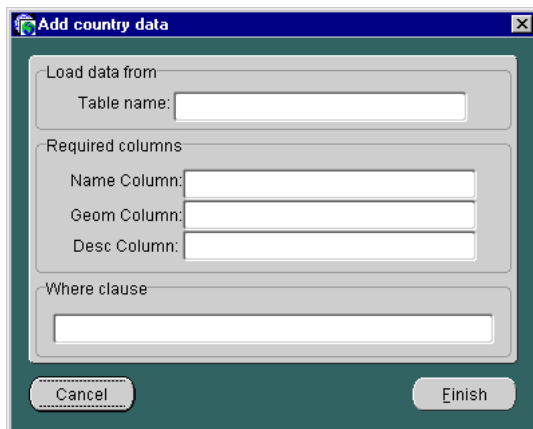
Field	Description
Table Name	The name of the table where the data is located.
Name Column	The name column of the table where the data is located.
Geometry Column	The geometry column of the table where the data is located.
Desc. Column	The description column of the table where the data is located.

4.3.6.2.1 Adding Data Under a Continent

To add data under a continent:

1. Right click a continent node and select Add Country. The Add Country Data form appears.

Figure 4–8 The Add Country Data Form



2. In the Add Country Data dialog, enter the table name where your data is located.

3. In the Required Columns portion of the Add Country Data form, specify the Name Column, Geometry Column, and Description Columns of the table where the data is located.
4. In the Where Clause field, specify a WHERE clause for the SQL statement. For example, if you are loading the first ten rows of your table, you can specify `rownum<11` as the WHERE clause.

Note: This is an optional step. If you do not specify a WHERE clause, the Region Modeling tool loads all rows from the specified table.

5. Click Finish to complete the country SDR.

4.3.6.2.2 Adding Data Under a Country

To add data under a country:

1. Right-click the country node for which you want to load data.
2. If that country does not have existing data, a form with the options to add state, postal code, city and county appears.
3. Enter the appropriate information.
4. Click Finish to complete the SDR.

Note: If the country has data for state, then you can add only state data. If the country does not have state data but instead has postal code, city, or country data directly under it, then you can add data for postal code, city, or country.

4.3.6.2.3 Adding State Data Under a Country

1. Right-click a country node for which you want to add state data.
2. Select Add state. The Add State Data form appears.

Figure 4–9 The Add State Data Form

3. In the Required Columns portion of the Add State Data form, specify the Name Column, Geometry Column, and Description Columns of the table where the data is located.
4. In the Where Clause field, specify a WHERE clause for the SQL statement. For example, if you are loading the first ten rows of your table, you can specify `rownum<11` as the WHERE clause.

Note: This is an optional step. If you do not specify a WHERE clause, the Region Modeling tool loads all rows from the specified table.

5. Click Finish to complete the state SDR.

4.3.6.2.4 Adding Data for the Postal Code, County, or City

To add data for the postal code, county, or city:

1. Right-click the Postal Code, City, or County folders. The menu listing options to add data for postal code, county or city appears.
2. From the menu, select folder to which you want to add data. The form for the Add postal code, Add county, or Add city menu options appears.
3. Enter the table name where your data is located.

4. In the Required Columns portion, specify the Name Column, Geometry Column, and Description Columns of the table where the data is located.
5. In the Where Clause box, specify a WHERE clause for the SQL statement. For example, if you are loading the first ten rows of your table, you can specify `rownum<11` as the WHERE clause.

Note: This is an optional step. If you do not specify a WHERE clause, the Region Modeling tool loads all rows from the specified table.

6. Click Finish to complete the SDR.

4.3.7 Deleting a System-Defined Region

To delete an SDR:

1. Right-click the SDR in the left frame of the Region Modeling Tool.
2. Select Delete.

Note: You can only delete SDRs in Standalone mode.

4.4 Creating Location-Based Master Services and Folders

You use the Region Modeling Tool in its browse mode when you make a service or a folder location-based by assigning a region. Folders and services created using the Region Modeling Tool are visibility location-based. Users access these services at specific locations.

In addition, you can use the Region Modeling Tool to create UDRs in the browse mode. For more information on creating UDRs, see [Section 4.3.2](#).

4.4.1 Accessing the Region Modeling Tool in the Browse Mode

You can access the Region Modeling Tool in its browse mode through the Service Designer or through the Master Service Creation Wizard.

In the Service Designer, you access the Region Modeling tool by selecting the Location Dependent check box in the either the General tab (used for modifying a folder or service in the Wireless Edition repository tree), or the *Create New Services* or the *Create New Folders* forms (invoked by right-clicking the Wireless Edition

repository) and then by clicking the Browse button. For more information on modifying a folder or a master service, see [Section 3.2.3, "Modifying Objects"](#) in [Chapter 3, "Wireless Edition Services"](#).

In the Master Service Creation Wizard, you access the Region Modeling Tool by selecting the Browse button in the *Creating New Folder* and *Setting Service Properties* screens.

4.4.1.1 Modifying a Folder or Service to be Location-Based

To assign a region to a folder or to a service from the Service Designer:

1. In the Wireless Edition repository tree, click the folder or service to which you wish to assign a region. The General tab appears.
2. In the General tab, click the Location Dependent check box. The Browse button appears.
3. Click the Browse Button. The Region Modeling Tool appears.
4. Right-click the region or regions that you wish to assign to the service or folder. Click Select.

Note: You can select multiple regions using <CTRL> + click.

5. The General tab reappears, showing the Area ID of the selected region.

4.4.1.2 Using the Service Designer to Create a Location-Based Folder or Service

To create a location-based folder or service from the Service Designer:

1. In the Wireless Edition repository tree, right-click the folder or service to which you wish to assign a region.
2. In the *Create New Folders* or *Create New Services* screens, click the Location Dependent check box. The Browse button appears.
3. Click the Browse Button. The Region Modeling Tool appears.
4. Right-click the region or regions that you wish to assign to the service or folder. Click Select.

Note: You can select multiple regions using <CTRL> + click.

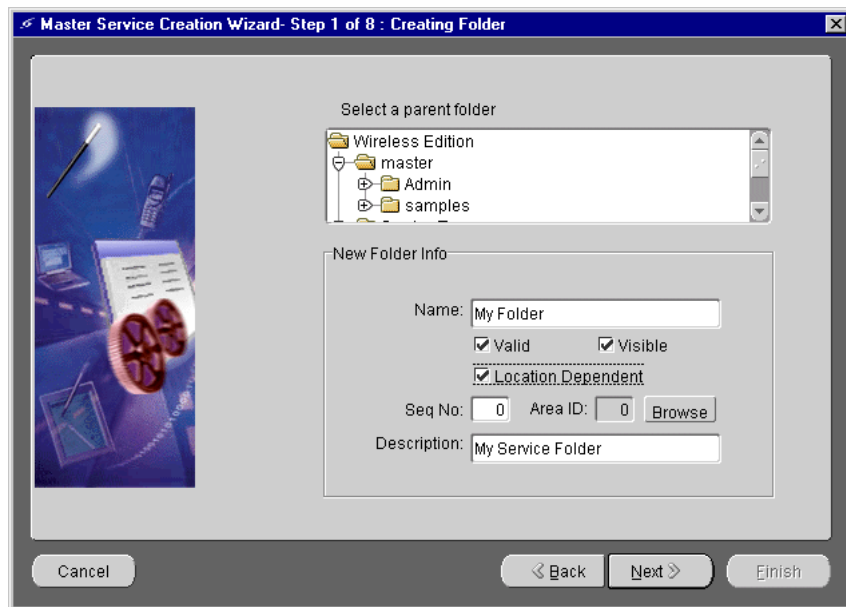
5. The *Create New Folders* or *Create New Services* screen reappears, showing the Area ID of the selected region.

4.4.1.3 Using the Master Service Creation Wizard to Create a Location-Based Service or Folder

To access the Region Modeling Tool from either the Create the Master Service Creation Wizard:

1. In either the *Creating New Folder* and *Setting Service Properties* screens, click the Browse button. The Region Modeling Tool appears.

Figure 4–10 The Creating Folder Screen of the Master Service Creation Wizard



2. In the left frame, select the region node to which you wish to assign a service. For example, from the SDR hierarchy, expand the North America node to the California node, and then expand to the San Francisco node.
3. Right click the region or regions (for example, San Francisco).

Note: You can select multiple regions using <CTRL> + click.

4. Click Select. The Creating Folder screen reappears, showing the Area ID of the selected node (for example, 8686, the Area ID of San Francisco). A location has been assigned to the service folder.

4.5 Creating Location-Based Services with Location-Specific Content

In addition to services that are visible only at certain locations, the Wireless Edition edition also supports location-based services that generate content specific to the current location of the Wireless Edition user. For example, a user located in San Francisco would receive content specific to the San Francisco area, such as a listing of restaurants or business located in San Francisco. These services are supported through customized adapter implementation and location-aware runtime support.

To create a content-specific location-dependent service, you must access the current subscriber's location through the Runtime API or through derived location-related system parameters.

4.5.1 Accessing the Current Subscriber's Location Using the Runtime API

From a customized Java adapter, access the subscriber's current location by using either of the following methods:

```
Request.getCurrentLocation()
Session.getCurrentLocation()
```

Both of these methods return the geocoded physical location point. If there is no request-specific location information, the Runtime API returns the current location information, if any.

4.5.2 Accessing the Current Subscriber's Location Using Derived Location-Related System Parameters

You can use derived location-related system parameters if you pass the location information to a built-in adapter, such as the WebIntegration adapter, or the SQL adapter. The supported derived location-related system parameters include:

Table 4-4 The Derived Location-Related System Parameters

Parameters	Description
<code>_Longitude</code>	The longitude component of the caller's current location.
<code>_Latitude</code>	The latitude component of the caller's current location.

Parameters	Description
_Postalcode	The postal code from which the caller is initiating the request.
_State	The state from which the caller is initiating the request.
_Country	The country from which the caller is initiating the request.

To enable the passing of location information to the existing adapter, you must add the corresponding parameters to the service and make them mandatory. For example, if you wish to pass the postal code information of the current caller to the router service, you must add the _Postalcode parameter to the router service and make it mandatory. The Runtime fills in the value for the derived system parameters with the mandatory values that are missing.

Note: A content-specific location-based service is not always a location-dependent service that is visible only in a specific area. If a content-specific location-based service is not visible only at a certain location, you do not need to select the Location Dependent check box.

Managing Users and Groups

This document describes how you can create and modify Oracle9i Application Server Wireless Edition users and user groups. Each section of this document presents a different topic. These sections include:

- [Section 5.1, "Overview"](#)
- [Section 5.2, "Managing Groups Using the Service Designer"](#)
- [Section 5.3, "Managing Groups and Users Using the Personalization Portal"](#)
- [Section 5.4, "Integrating Users with Existing Provisioning Systems"](#)

5.1 Overview

The Wireless Edition provides administrative user and group management and configuration functions in both the Service Designer and the Personalization Portal.

5.2 Managing Groups Using the Service Designer

This section describes how you can create, modify, or delete user groups using the Wireless Edition Service Designer:

- [Section 5.2.1, "Creating User Groups Using the Service Designer"](#)
- [Section 5.2.2, "Deleting Groups Using the Service Designer"](#)
- [Section 5.2.3, "Adding Users to a Group Using the Service Designer"](#)
- [Section 5.2.4, "Adding Services to a Group Using the Service Designer"](#)
- [Section 5.2.5, "Removing Users from a Group Using the Service Designer"](#)
- [Section 5.2.6, "Removing Services from a Group Using the Service Designer"](#)

5.2.1 Creating User Groups Using the Service Designer

The Wireless Edition users can be organized into groups. Each Wireless Edition user can belong to several groups. Services can also be assigned to these user groups and can be made available to all users in that group. To create a group:

1. Right-click Groups in the Wireless Edition repository tree view.
2. Click Create New Group to display the Create New Group form.

Figure 5–1 Create New Group Form

The Create New Group form includes the following parameters:

Table 5–1 Parameters of the Create New Group Form

Parameter	Value
Name	A unique name for the group.
Available Users: Name	A string for querying the user's name from the list of available users.
Available Users: Disp. Name	A string for querying the user's display name from the list of available users.

Parameter	Value
Group Members: Name	A string for querying the user's name from the group members.
Group Members: Disp. Name	A string for querying the user's display name from the list of selected users.

3. Enter the name of the new user group in the Name field at the top of the screen.
4. In the Available Users section of the Create New Group form, enter the name of the user you wish to add in the Name field or enter the user's display name.
5. Click Find. To display all available users, leave the Name and Disp. Name fields blank and click Find. Results of the search criteria are displayed by user name and display name in the Available Users section.

Note: Searching the Name and Disp. Name fields in the Available Users section returns only users who are not assigned to any group.

6. Select the user you want to add to the group. You can select multiple users by holding the CONTROL key.
7. Click the right arrow (>) to move users from the Available Users section to the Group Members section. Clicking the double right arrows (>>) adds all available users to the new group.
8. Click Next.

After creating the group, you can make services available to the group members by assigning service folders to the group. To assign a service to a group:

1. If not already selected, select the group from the Wireless Edition tree view.
2. Click the Service Folders tab.
3. Expand, if necessary, the root service folders shown to display the folder you want to make available to the group.
4. Select the folder you want to assign to the group. Holding the CONTROL key enables you to select multiple folders.

5. Click the right arrow (>) to move folders to the Selected Service Folders field. Clicking the double right arrows (>>) moves all folders to the Selected Services Folder field.
6. Click Finish.

5.2.2 Deleting Groups Using the Service Designer

To delete a group:

1. Expand the Groups folder in the Wireless Edition repository tree view.
2. Select the group you want to delete by clicking the right mouse button.
3. Click Delete.
4. Click Yes in the Confirm Delete dialog.

5.2.3 Adding Users to a Group Using the Service Designer

To add a user to a group:

1. Expand the Groups folder in the Wireless Edition repository tree view.
2. Select the group to which you want to add users. The Group Properties form appears in the right frame. Select the General tab, if it does not already appear.
3. In the Available Users section, enter the name of the user you wish to add in the Name field or enter the user's display name.
4. Click Find. To display all available users, leave the Name and Disp. Name fields blank and click Find. Results of the search criteria are displayed by user name and display name in the Available Users section.
5. Select the user you want to add to the group. You can select multiple users by holding the CONTROL key.
6. Click the right arrow (>) to move users from the Available Users section to the Group Members section. Clicking the double right arrows (>>) adds all available users to the group.
7. Click Apply.

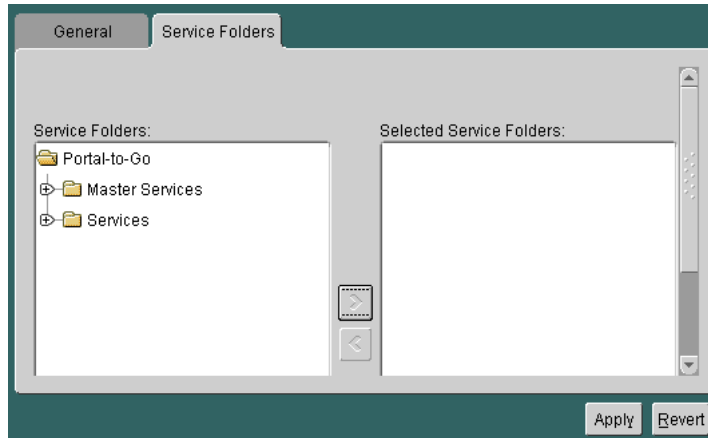
5.2.4 Adding Services to a Group Using the Service Designer

To make services available to a group:

1. Expand the Groups folder in the Wireless Edition repository tree view.

2. Select the group name to which you want to add services.
3. Click the Service Folders tab to display the Service Folders panel.

Figure 5–2 The Service Folders Tab



4. Under Service Folders, expand the folders, if necessary, to display the folder that you want to assign to the group. Select the folder. You can select multiple folders by holding the CONTROL key.
5. Click the right arrow (>) button to move the folder to the Selected Service Folders field.
6. Click Apply.

5.2.5 Removing Users from a Group Using the Service Designer

To remove users from a group:

1. Expand the Groups folder in the Wireless Edition repository tree view to display the available groups.
2. Select the group from which you want to remove users. The group properties form appears in the right frame. Select the General tab, if it does not already appear.
3. In the Available Users section, enter the name of the user you wish to remove from the group in the Name field or enter the user's display name.

4. Click Find. To display all available users, leave the Name and Disp. Name fields blank and click Find. Results of the search criteria are displayed by user name and display name in the Available Users section.
5. Select the user you want to remove from the group. Hold the CONTROL key to select multiple users for removal.
6. Click the left arrow (<) to move users from the Group Members section to the Available Users section. Clicking the double left arrows (<<) removes all users from the group.
7. Click Apply.

5.2.6 Removing Services from a Group Using the Service Designer

To remove services from a group:

1. Expand the Groups folder in the Wireless Edition repository tree view to display the available groups.
2. Select the group from which you want to remove services. The group properties form appears in the right frame.
3. Click the Folders tab.
4. Under Selected Service Folders, select the folder you want to remove. You can select multiple folders by holding the CONTROL key.
5. Click the left arrow (<) to remove the folder from the Selected Service Folders field. Clicking the double left arrows (<<) removes all folders from the group.
6. Click Apply.

5.3 Managing Groups and Users Using the Personalization Portal

The Personalization Portal provides a browser interface that administrators use to both configure and create users.

The Personalization Portal organizes groups, users, and services as a hierarchy of folders. For example:

- A group is a folder containing users.
- A user is a folder containing sub-folders and services.
- A service is an alias for a master service created using Service Designer.

In addition to managing users and groups, administrators can use the Personalization Portal to create such objects as alerts and location marks. Administrators can also use the Personalization Portal to perform service assignment, subfolder creation, URL bookmark association, and manipulation of other Wireless Edition repository objects. For more information on these uses of the Personalization Portal, see [Chapter 6, "Personalization from Desktops"](#).

Administrators can also update repository objects by adding pages to them or creating customized pages for them. The Personalization Portal framework and interface are designed to be customizable. For more information on customizing these pages, see the Oracle9i Application Server Wireless Edition Developers Guide.

5.3.1 Accessing the Personalization Portal

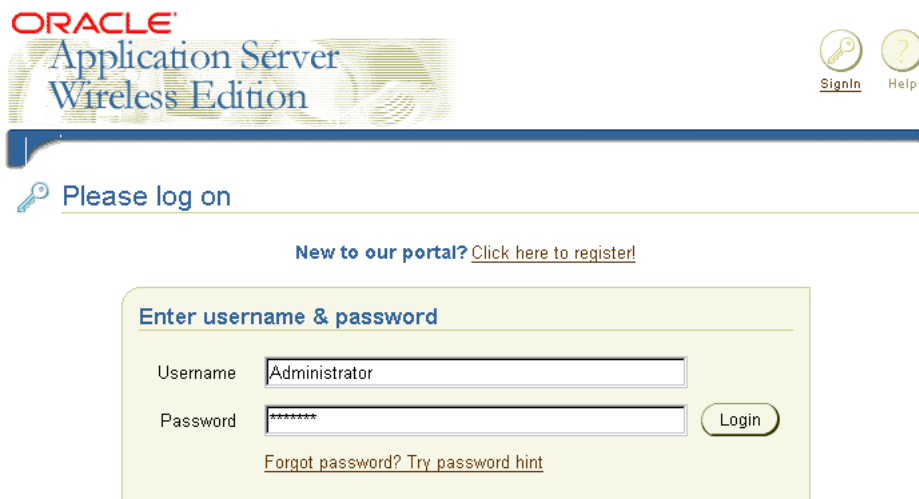
Before using the Personalization Portal, you must first use your browser to access the portal. From a Web browser, enter the following URL:

```
http://9iASWEServer.domain/portal/Login.jsp
```

Note: The default listener port for the Oracle9i Application Server is 7777. The URL for this port is `http://9iASWEServer.domain:7777`. Refer to the Oracle9i Application Server documentation to change the HTTP listener to port 80.

After you enter the URL, the Service screen of the Personalization tab appears.

Figure 5–3 The Services screen of the Personalization Tab



The Service tab presents the following login information for the Personalization Portal:

Table 5–2 Login Elements of the Service Tab of the Personalization Portal

Elements	Description
User Name	The name of the Personalization Portal user.
Password	The password of the Personalization Portal User.
Password Hint	Clicking the "Forget Password?" link invokes a prompt that helps users remember their passwords.
Login	Clicking this button gives authorized users access to the Personalization Portal.

5.3.2 Logging in as an Administrator

To log into the Personalization Portal as an Administrator:

1. Enter Administrator as the user name.

2. Enter manager as the password.

Note: Both the user name and password are case-sensitive.

The Administration tab of the Personalization Portal appears with the User page selected.

Figure 5-4 The User Page of the Personalization Portal

The screenshot shows the Oracle Application Server Wireless Edition Administration interface. At the top, there is a navigation bar with tabs for Home, Personalization, and Administration. Below the navigation bar, there are links for User, New User, and Group. The main content area is titled 'Users' and contains a 'Search Users' section with input fields for 'User Name' and 'Display Name', and a 'Search' button. Below the search section is a 'Found User List' table with columns for 'Select', 'User Name', 'Display Name', 'Group', and 'User Home'. The table is currently empty. To the right of the search and list sections is a 'Quick Tutorial' sidebar with a list of five steps for managing users.

Search Users

User Name

Display Name

Found User List

⊖ Previous 1-10 Next 10 ⊕

Select	User Name	Display Name	Group	User Home

⊖ Previous 1-10 Next 10 ⊕

Quick Tutorial

This page is to create/view/edit/delete users by an administrator.

1. Enter user name and/or display name and press Search button to search users.
2. Select one or more users and press Delete button to delete the selected users.
3. Select one or more users and press AssignGroup button to assign groups for the selected users.
4. Click on a user name link to modify the user profile.
5. Click on a user home link to view/customize the user services.

5.3.3 Using the Administration Tab

The Administration tab enables the creation and management of user accounts, including group membership, user profile, and the services associated with that user. The Administration tab contains the following screens:

The User Page

The User page enables administrators to search for users using both user name and the display name of a user.

Note: No users are listed in the User screen when it first displays. Use the Find User feature to display a list of users.

The search results display in the Found User List. The Found User List enables administrators to delete a user, edit the profile of a user, or assign the user to groups.

Note: The Found User List displays up to ten users at a time.

The New User Page

The New User page enables administrators to create a new user account and assign passwords, privileges, and services. For more information on finding users, see [Section 5.3.3.2](#).

The Group Page

The Group page enables the creation and management of groups including user membership, service association, and editing group properties.

5.3.3.1 Finding Users

The User page contains the Users screen. The Users screen contains User Name, Display Name fields that Administrators use to find current Wireless Edition users. To find users, administrators enter the user's name in the User Name field and click the Search button. The list of users corresponding to the search criteria appear in the Found User List.

Administrators can narrow their searches using the User Name field by adding the display name of the user in the Display Name field and then clicking the Search button. Conversely, administrators can display all current users in the Found User List by first entering a wildcard (* or %) in the User Name field and then clicking the Search button. In addition, wildcards (*or %) can be inserted in place of characters, or can replace all characters from the current position to the end of the string. The question mark (?) can be used to replace a single character.

The Found Users List is comprised of the following:

Table 5–3 Elements of the Found User List

Label	Definition
Select	Selecting this check box and clicking the Delete button removes a user. Selecting this check box and clicking the AssignGroup button invokes the Assign Group to User screen, where an administrator can reassign a user's group membership. Multiple users can be selected for group deletion or for batch assignment. For more information on assigning group membership, see Section 5.2.3
User Name	The name of the user. Clicking this link invokes the User Profile screen, where administrators can edit a user's properties. For more information on editing user properties, see Section 5.3.3.3 .
Display Name	The display name of the user.
Group	The Group to which the user has been assigned.
User Home	A link to the user's service tree.

5.3.3.2 Creating Users

The New User page contains a set of parameters that administrators set to create and configure new users.

Figure 5–5 The New User Page

Users

New User

User Name	Display Name	Group	User Home

Create a New User

* User Name

Display Name

* New Password

* Password Confirmation

Password Hint

Group

Language

Country

Enabled

Administrator Privilege

Designer Privilege

Use Auto Location

The New User page contains the following parameters.

Table 5–4 Parameters for the New User Page

Parameter	Description
User Name	The name of the user. Note: this is case-sensitive.
Display Name	The display name of the user.
New Password	The user’s password. Note: the password is case-sensitive.
Password Confirmation	The user’s password.
Password Hint	A prompt for the user’s password.

Parameter	Description
Group	A list box entry appears for each group which can include this user. You can assign a user to multiple groups by using <CTRL> + click in the Group list box.
Language	A drop-down list displaying the supported national languages.
Country	A drop-down displaying the options for user's country.
Enabled	Selecting this check box enables the new users to log in. Leaving this check box empty prevents a new user from logging in.
Administrator Privilege	Selecting this check box assigns a user to the Administrator role. The Administrator role enables a user to change the Wireless Edition configuration and can start and stop any component. In addition, users with the Administrator role can update privileges on any part of the Wireless Edition repository and can install and remove any service.
Designer Privilege	Selecting this check box assigns a user to the Designer role. Users assigned the Designer role can create a URL service and can test and deploy services.
Use Auto Location	This check box enables the user's location be reported to the server for personalized content from location-dependent services.

To create a new user:

1. Select the New User page.
2. Complete the parameters as follows:
 - a. Enter the user name in the User Name field.
 - b. Enter the display name for the user in the Display Name field.
 - c. Enter a password for the user in the Password field.
 - d. Enter the password again in the Confirm Password field.
 - e. Enter a password hint that is meaningful to the user in the Password Hint field.
 - f. From the Group list box, select a group (or groups) for the user.
 - g. Select a language from the Language list box.

- h. Select a country from the Country list box.
- i. Select the Enabled check box to allow the end user to log into the Personalization Portal and the phone portal.
- j. Select from among the Administrator Privilege or Designer Privilege check boxes to assign a role to the user. This is an optional step.

Note: Leaving the Administrator Privilege and Designer Privilege check boxes empty (the default setting) limits the user’s access to that of an end user. Users with the end user role have access only to the objects they own.

- k. Select the Auto Location check box to get personalized results from location-based services. This is an optional step.
- l. Click create to complete the creation of the user. The new user appears at the top of the screen, confirming the creation of the new user.

Figure 5–6 New User Confirmation

Users

New User

User Name	Display Name	Group	User Home
Scott Tiger	Scott	Users;	/Users Home/Scott Tiger

Clicking the links under the User Name invokes the Profile page of the Personalization tab, where administrators can edit user properties. Clicking the links under User Home invokes the Service page of the Personalization tab, where administrators can personalize the new user’s services, run the services and create alerts for the new user. For more information on editing user properties, see [Section 5.3.3.3](#).

5.3.3.3 Editing User Properties

The Found Users List enables administrators to edit user profiles and edit services available to a user. In addition, the Found Users List enables administrators to delete users from the Wireless Edition repository or assign users to groups.

Figure 5–7 The Found User List

Found User List

For Selected Users Delete AssignGroup

⊖ Previous 1-10 Next 10 ⊕

Select	User Name	Display Name	Group	User Home
<input type="checkbox"/>	Administrator		Admin;	/Administrator
<input type="checkbox"/>	Scott Tiger	Scott	Users;	/Users Home/Scott Tiger
<input type="checkbox"/>	jshi	Jimmy Shi	Admin;Users;	/Users Home/jpsi

To edit a user’s profile, click the desired link under the User Name label. The Profile screen of the Personalization tab appears. The parameters of the Profile screen are populated by the values entered in the New User page.

Figure 5–8 The User Profile Page

User Profile

* User Name

Display Name

New Password

Password Confirmation

Password Hint

Group

Language

Country

Default Alert Address

Default LocationMark

Enabled

Administrator Privilege

Designer Privilege

Use Auto Location

Cancel Apply

The Profile page has the following parameters.

Table 5–5 Parameters for the User Profile Page

Parameter	Description
User Name	The name of the user. Note: this is case-sensitive.
Display Name	The display name of the user.
New Password	The user’s password. Note: the password is case-sensitive.
Password Confirmation	The user’s password.
Password Hint	A prompt for the user’s password.
Group	A list box entry appears for each group which can include this user.
Language	A drop-down list displaying the supported national languages.
Country	A drop-down displaying the options for user’s country.
Default Alert Address	The address that alerts are sent to.
Default LocationMark	A drop-down list that displays the options for a default landmark.
Enabled	Selecting this check box enables new users to log in. Leaving this check box empty prevents users from logging in.
Administrator Privilege	Selecting this check box assigns a user to the Administrator role. The Administrator role enables a user to change the Wireless Edition configuration and can start and stop any component. In addition, users with the Administrator role can update privileges on any part of the Wireless Edition repository and can install and remove any service.
Designer Privilege	Selecting this check box assigns a user to the Designer role. Users assigned the Designer role can create URL services, and can test and deploy any service.
Use Auto Location	This check box enables the user’s location to be reported to the server for personalized content from a location-dependent service.

To edit a user profile:

1. Click the desired link under the Names label of the Found Users List. The Profile page appears.

2. Change the following parameters as needed:
 - a. Enter a new user name in the User Name field.
 - b. Enter a new display name for the user in the Display Name field.
 - c. Enter a new password for the user in the Password field.
 - d. Enter the new password again in the Confirm Password field.
 - e. Enter a password hint that is meaningful to the user in the Password Hint field.
 - f. From the Group list box, select a new group for the user.
 - g. Select a language from the Language list box.
 - h. Select a country from the Country list box.
 - i. Select a default alert address from the Default Alert Address list box.
 - j. Select a default locationmark address from the Default LocationMark list box.
 - k. Select the Enabled check box to allow the end user to see the available services.
 - l. Select from among the Administrator Privilege or Designer Privilege check boxes to assign a role to the user. This is an optional step.

Note: Leaving the Administrator Privilege and Designer Privilege check boxes empty (the default setting) limits the user's access to that of an end user. Users with the end-user role have access only to the objects they own.

- m. Select the Auto Location check box to report the user's location to the server for personalized content from a location-dependent service. This is an optional step.
3. Click Apply. The Users page appears, with any changes pertinent to the labels in the Found Users List (for example, the user name) visible.

5.3.3.4 Assigning a User to a Group

The Found User List also enables administrators to assign users to groups by giving them access to the Assign Group to User page

Figure 5–9 The Assign Group to User page

[Administration](#) > User

Assign Group to User

Add to Following Groups
 Remove from Following Groups
 Set Only to Following Groups

Available Groups

Admin
 DocTest
 MySampleGroup
 Users

The Assign Group to User screen contains the following elements:

Table 5–6 Elements of the Assign Group to User Screen

Element	Description
Add to Following Groups	Selecting this radio button enables administrators to add selected users to the group or groups selected from the Available Groups list box. Users retain their membership to groups to which they already belong.
Remove from Following Groups	Selecting this radio button enables administrators to remove a user from a group or groups selected from the Available Groups list box. Users retain their membership to groups to which they already belong.
Set Only to Following Groups	Selecting this radio button enables administrators to remove all the users currently belonging to a group and to add users to selected groups.
Available Groups	A list box of available groups in the Wireless Edition repository. Use <SHIFT> + click to select multiple groups.

To assign a user to a group:

1. Select the check box next to the desired user.
2. Click the AssignGroup button. The Assign Group to User screen page.
3. Select the appropriate group assignment option radio box.

4. Select a group or groups from the Available Groups list box.
5. Click Apply. The Users screen reappears, with the new group membership visible.

Note: Users retain their membership to groups to which they already belong. For example, if a user belongs to Group 1 and is assigned to Group 2, that user will retain membership to Group 1. However, if an administrator uses the Set Only to Following Groups function to assign this Group 1 user to Group 2, the user is no longer a member of Group 1 and now belongs exclusively to Group 2.

5.3.4 Managing Groups Using the Personalization Portal

Administrators use the Group page to create and manage Wireless Edition groups. The Groups page is divided into two sections: the Groups List section and the Create a New Group section. The Groups List section enables administrators to manage groups by enabling such functions services and user assignment while the Create a New Group section enables administrators to create new groups in the Wireless Edition repository.

5.3.4.1 Creating a Group Using the Personalization Portal

To create a group using the Personalization Portal:

1. In the Create a Group portion of the Groups page, enter the name of the group you want to create in the New Group Name field.
2. Click Create. The new group appears in the Group Name section.

5.3.4.2 Managing Groups

The Groups List section of the Groups page enables administrators to delete groups, assign users to groups, and to assign services to groups.

Figure 5–10 The Groups Page

Groups

Group List

For Selected Groups		
		<input type="button" value="Delete"/> <input type="button" value="UserList"/> <input type="button" value="SetServices"/> <input type="button" value="Apply"/>
Select	Group Name	Group Service
<input checked="" type="radio"/>	<input type="text" value="Admin"/>	/Administrator/master/samples;
<input type="radio"/>	<input type="text" value="Users"/>	/Users;

Create a New Group

* New Group Name

The section contains the following elements.

Table 5–7 Elements in the Group List Section of the Groups Page

Element	Description
Select	Select this radio button selects the adjacent Group for editing.
Group Name	The name of the Group. This field can be edited. Click Apply after editing.
Group Service	The service or services assigned to the group. The services are presented as a hyperlink. Clicking a link displays the Group Service Subscription Page, where administrators can edit the services available to a group.

Administrators use the Delete, Userlist, SetServices, and Apply buttons with the Select radio box to manage a group’s users and service subscriptions.

5.3.4.3 Deleting a Group

To delete a group:

1. Select the radio box next to the group that you want to remove from the Wireless Edition repository.

2. Click the Delete button.

5.3.4.4 Adding Users to a Group

To add users to a group:

1. Select the radio box next to the group to which you want to add users.
2. Click UserList. The Found User List appears.
3. Enter the name of the user you want to add, or enter * or % to return a list of all users. The User page from the Administration tab appears.

Figure 5–11 The Found User List

Found User List

For Selected Users Delete AssignGroup

Previous 1-10 Next 10

Select	User Name	Display Name	Group	User Home
<input type="checkbox"/>	Administrator		Admin;	/Administrator
<input type="checkbox"/>	Administrator1		Admin;	/Users Home/Administrator1
<input checked="" type="checkbox"/>	Betty	Betty	Users;	/Users Home/Betty

4. Click AssignGroup. The Assign Group to User page appears.

Figure 5–12 The Assign Group to User Page

Assign Group to User

Add to Following Groups
 Remove from Following Groups
 Set Only to Following Groups

Available Groups

Users
 group1
 group10
 group12
 group5

Cancel Apply

5. Select the Add to Following Groups radio box.
6. Select the group from the list box.
7. Click Apply.

5.3.4.5 Assigning Services to a Group

To assign services to a group:

1. Select the radio box next to the group to which you want to add services (For example, Group 12).
2. Click SetServices. The Select Group Service for Group page appears. For this example, the Select Group Service for Group screen displays the services available to Group 12.

Figure 5–13 The Select Service for Group Page



3. Select the check box (or check boxes) adjacent to the desired service.

Note: The services are displayed as branches off of the three main entries, Administrator, Users Home, and Master. Selecting a higher-level branch selects everything below it. Typically, end-users have only access to Users Home.

4. Click the Apply button. The Group List page reappears with the services assigned to the group visible as a hyperlink. Clicking this hyperlink displays the service tree that belongs to the group, enabling administrators to personalize the group's services.

Note: For information on editing services, see [Chapter 6, "Personalization from Desktops"](#).

5.4 Integrating Users with Existing Provisioning Systems

You can integrate your repository with an existing provisioning system using one of the following:

1. You can manage users with the Wireless Edition provisioning adapter. The provisioning adapter supports the Wireless Edition services that create, update, and delete users at runtime.
2. Use the `LoadXml` utility to import users from a flat XML file that conforms to the Repository DTD. This utility enables you to create and update user information. For more information on the `LoadXML` utility, see [Chapter 2](#).
3. You can look up users at runtime by using the `UserAuthenticationHook` and the `ProvisioningHook` interfaces. This is similar to using the `LoadXML` utility, but with the hook interfaces you can import users one-by-one. The `LoadXML` utility operates on the entire repository. The following sections describe these interfaces.

5.4.1 ProvisioningHook Interface

You use the `ProvisioningHook` interface to integrate an external provisioning system with the Wireless Edition. The Wireless Edition locates the provisioning class to use in the **System.properties** file as the value of the `locator.provisioning.class` parameter. The implementing class must be implemented as a singleton class. This interface has the following methods:

- `getRootFolder()`: Creates a root folder for a user.
- `getUsersRootFolder()`: Gets the common root folder for all users.

5.4.2 UserAuthenticationHook Interface

The Wireless Edition calls the user authentication class when it cannot locate a user. The class must be implemented as a singleton class. This interface has the following methods:

- `authenticate()`: Authenticates a Wireless Edition user.
- `checkUser()`: Checks if the user exists, and if it does, sets appropriate attributes.

Personalization from Desktops

This document describes how to use the Personalization Portal from a browser interface. Each section of this document presents a different topic. These sections include:

- [Section 6.1, "Overview"](#)
- [Section 6.2, "Logging in to Personalize the Portal"](#)
- [Section 6.3, "Logging In as a User"](#)
- [Section 6.4, "Activating a Service for a User"](#)
- [Section 6.5, "Editing a Folder"](#)
- [Section 6.6, "Creating a Subfolder"](#)
- [Section 6.7, "Creating a Bookmark"](#)
- [Section 6.8, "Creating Alerts"](#)
- [Section 6.9, "Managing Alert Addresses"](#)
- [Section 6.10, "Creating a LocationMark"](#)
- [Section 6.11, "Editing a LocationMark"](#)
- [Section 6.12, "Changing the Default Status of a LocationMark"](#)
- [Section 6.13, "Deleting a LocationMark"](#)
- [Section 6.14, "Editing a User Profile"](#)

6.1 Overview

You can use access the Personalization Portal from a browser interface to both customize the Wireless Edition portal and assign Wireless Edition mobile services. In addition, you can create the objects that Wireless Edition uses to perform its functions.

The Personalization Portal operates on a folder model. For example:

- A group is a folder containing users and services.
- A user is a folder containing sub-folders, services, location marks, bookmarks, and alerts.
- A service is an alias for a master service created using Service Designer or by a user with designer privileges using the Personalization Portal.

Objects which can be created by a user are users, folders, alerts, and locationmarks. Other operations that can be performed using this interface include service assignment, subfolder creation, and URL bookmark association, as well as manipulation of other Wireless Edition repository objects.

6.2 Logging in to Personalize the Portal

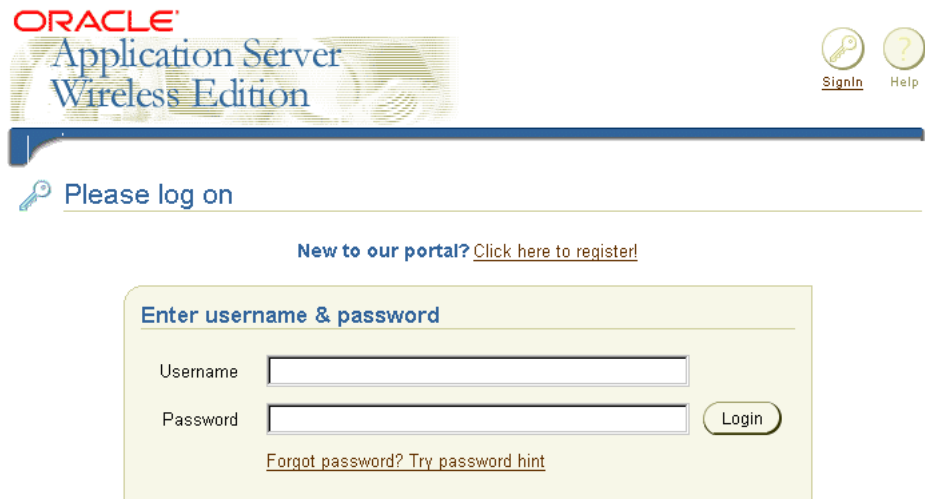
Before using the Personalization Portal, you must first use your browser to access the portal. From a Web browser, enter the following URL:

```
http://9iASWEServer.domain/portal/Login.jsp
```

Note: The default listener port for the Oracle9i Application Server is 7777. The URL for this port is `http://9iASWEServer.domain:7777`. Refer to the Oracle9i Application Server documentation to change the HTTP listener to port 80.

After you enter the URL, the Service screen of the Personalization tab appears.

Figure 6–1 The Login Screen of the Personalization Portal



The Service tab presents the following login information for the Personalization Portal:

Table 6–1 Login Elements of the Service Tab of the Personalization Portal

Elements	Description
User Name	The name of the Personalization Portal user.
Password	The password of the Personalization Portal User.
Password Hint	Clicking the "Forgot Password?" link invokes a prompt that helps users remember their passwords.
Login	Clicking this button gives authorized users access to the Personalization Portal.

A default page appears when you log in as a user with Home and Personalization tabs in the upper-right corner.

Figure 6–2 The Service Subscription Page



The Service page includes the Service, Alert, Address, LocationMark and Profile pages.

Table 6–2 The Personalization Portal Tabs

Element	Contents
Personalization	Displays the Service, Alert, Alert Address, Location Mark, and Profile tabs.
Home	A generic tab which can be easily rebranded. This page contains links to the Self-Registration page and the Login page.
Services	Displays the Service Subscription page. You can copy a service, edit, create, or delete an object. Designers can create, test, and post URL services on this page.

Element	Contents
Alerts	Displays the Alert List page. You can edit or delete alerts on this page.
Alert Address	Displays the Alert Address List page. You can update, delete, create, or change the default status of an Alert Address on this page.
LocationMark	Displays the LocationMark List page. You can create, delete, edit, or change the default status of a LocationMark on this page.
Profile	Displays the User Profile page. You can edit the user home folder profile on this page.

6.3 Logging In as a User

A newly created user has an empty home folder bearing the user name. New users are assigned to a default group unless created by an administrator. Only an administrator can assign a user to a group or groups.

Using the tabs under the Personalization tab, users can:

- Copy services.
- Create folders and bookmarks.
- Select alert addresses.
- Schedule the delivery of alerts.
- Create and edit location marks.
- Edit personal profile information.

6.4 Activating a Service for a User

The Service Subscription page is the default page when you log in as a user or when you click the Service tab. The services and folders that exist within a user's group are displayed as a service tree under the Group Services heading. The user services and folders display under the Personal Services heading.

Figure 6–3 The Personalization Portal Icons

Service Subscription

Group Services

- ▼ Finance
 - ▮ CNNFN
 - ▮ YahooFinance
- ☎ HelloAdapter
- ☎ Time
- ▼ Users
- ▼ News
 - ▮ CNN
 - ▮ Yahoo
- ▼ Portals
 - ▮ Yahoo

Table 6–3 Personalization Portal Icon Functions

Symbol	Function
Circle with down arrowhead	User, folder, or subfolder. Click the name to display.
Phone Icon	Service or adapter. Click the icon to activate.
Bookmark ribbon	Bookmark to an external URL. Click the ribbon to activate.

A user can use any service in the Group Services column. Services are activated by clicking on the symbol next to the service name. Users can personalize most services. A User personalizes services by copying the service and then modifying its input parameters to reflect that user’s preferences. For example, a user can modify a stock price-quote service by selecting the stock’s ticker symbol as an input parameter.

6.4.1 Copying or Editing a Service

Administrator level users can alter any service except a master service. Master services are created using the Service Designer and cannot be altered using the Edit Service option in the Personalization Portal. Non-administrator level users can only alter services that they own. A user owns a service if:

- The user is listed as the owner when the alias is created in the Service Designer.
- The user has made a personal copy of a service.

To copy a service:

1. Click a service under the Group Services column. The Copy Service page appears.

Figure 6–4 The Copy Service Page

Service | Alert | AlertAddress | LocationMark | Profile

Personalization > Service

Copy Service

Create Alert

Edit Service

* Name

Move to Folder

Input Arguments

Argument	Value
symbols	<input type="text" value="orc1"/>

Cancel Create

2. Edit the text in the Name field.

Note: Because the user home folder contains copies of the group services, the name in the Move to Folder field is the same name that you used to log into the Personalization Portal.

3. Alter any input parameters as needed. Leaving an input parameter text box blank means that the service prompts you for input when you activate it.
4. Click the Create button to create the copy of the service and return to the previous page. The new service appears under the Personal Services column and is selected.

5. Your folders and services appear as links with check boxes.
6. Selecting a check box and clicking on the Apply button makes that service visible or invisible.

Editing a service you own uses the same steps, except that you select the Edit a Service option and the page title reads Edit a Service.

6.4.1.1 Configuring Input Arguments and Conditions

These elements vary from service to service, and are established when an administrator creates a master service using the Service Designer. For more information on this, see [Chapter 3, "Wireless Edition Services"](#).

The Personalization Portal allows you to use these services and input parameters yourself, or copy certain services and pre-configure the input parameters, personalizing it for your use.

The following example is for a stock quote service.

The input argument for an alert linked to a stock price-quote service is the stock's ticker symbol.

Conditions for this service include four types of arguments, Change, Price, Volume, and Change Per. Each argument can be modified by a drop-down list which specifies Greater, Less, Equal, or Contains. Each argument can also have a value entered in the text box.

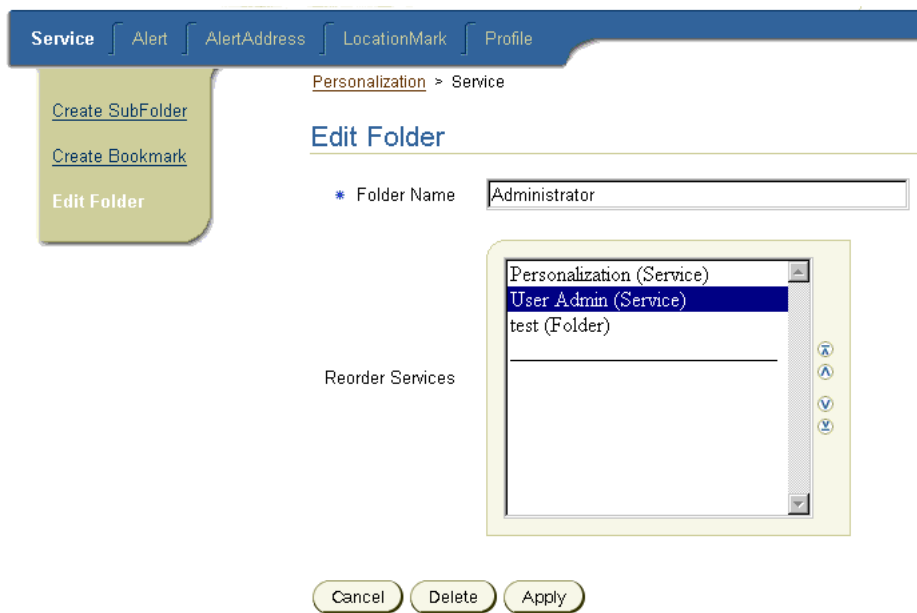
6.5 Editing a Folder

To edit a folder:

1. Click the Service tab.
2. Click any folder name to display the Edit Folder page.
3. The Edit Folder page displays the contents of the active folder, allows the creation of new subfolders and bookmarks using the Create Sub Folder and Create Bookmark links.

6.5.1 Reordering Folder Contents

You can modify the order in which subfolders and services appear using the Reorder Services list box in the center of the Edit Folder page.

Figure 6–5 The Reorder Services List Box

To reorder services:

1. Click the service or subfolder you wish to move and use the scroll arrows on the side of the list box to adjust its position.
2. Click the Apply button to complete the reordering to the user name folder and return to the Service Subscription page.

In addition to the apply button, the Edit Folder page contains a Delete button and a Cancel button. Clicking the Delete button deletes a service or a subfolder. Clicking the Cancel button suspends service or subfolder reordering and returns you to the Service Subscription page.

6.6 Creating a Subfolder

Subfolders can be created inside the user home folder or another subfolder and can contain services, bookmarks, locationmarks, and other folders. To create a subfolder:

1. Click the Service tab.
2. Click any folder, including the user home folder.
3. Click the Create Sub Folder link to display the Create Sub Folder page.
 - a. Enter the desired name in the Folder Name text box.
 - b. Select the location for the new subfolder using the Move to Folder drop-down list box. The default location is as a sub folder of the user home folder.

Clicking the Cancel button suspends the process and returns to the Service Subscription page. Clicking the Create button creates the subfolder in the appropriate folder and returns you to the Service Subscription page where the new subfolder is displayed. The new sub folder can be populated with services, bookmarks, and subfolders.

6.6.1 Moving Objects Between Folders

Objects can be moved from one folder to another within a user's home folder. To move an object between folders:

1. Click the object name to select it.
2. Use the Move to Folder drop-down list box to select the destination folder.
3. Click the Apply button. The object is moved to that folder and you are returned to the Service Subscription page.

6.7 Creating a Bookmark

A bookmark is a link to a URL. To create a bookmark:

1. Click the Service tab.
2. Click any folder including the user home folder.

Figure 6–6 The Create Bookmark Screen

Service | Alert | AlertAddress | LocationMark | Profile

Personalization > Service

Create SubFolder
Create Bookmark
Edit Folder

Create Bookmark

* Name

URL

Move to Folder

Cancel Create

3. Click the Create Bookmark link to display the Create Bookmark page.
 - a. Enter the bookmark name in the Name text box.
 - b. In the URL text box, enter the URL of the new bookmark (for example, `www.oracle.com`).
 - c. Use the Move to Folder drop-down list box which determines the location of the new bookmark.

Clicking the Cancel button aborts the process and returns to the Service Subscription page. Clicking the Create button creates the bookmark in the appropriate folder and returns to the Service Subscriptions page where the new bookmark is displayed.

6.8 Creating Alerts

An alert is a notification service that alerts a user to important information or events. You use to AlertAddress page to create an alert. To access the AlertAddress page:

1. Click the Service tab.
2. Click any service. The Edit Service Page appears.

3. Click the Create Alert link. The Create Alert page displays previously created alerts. For more information on managing previously created alerts, refer to [Section 6.9](#).

6.8.1 Setting Alert Parameters

To set alert parameters:

1. Enter the alert name in the Name text field.
2. Use the Alert Address drop-down box to select the destination for the alerts.
3. Set the frequency for the alert by selecting from the Alert Sent Every and Alert Start Time radio buttons, data entry fields and drop-down menus. For more information the frequency settings, see [Section 6.8.1.1](#) and [Section 6.8.1.2](#).
4. Click Create.

6.8.1.1 The Alert Sent Every Settings

The Alert Sent Every Settings determine the alert frequency. The top setting consists of a drop-down list for selecting a day of the week, weekday, weekend, or every day settings.

The bottom setting consists of a number field and a frequency drop-down list with a range of hour(s) to years(s). For example, you could schedule an alert to be delivered every hour, only on Monday, only on weekends, and so on.

6.8.1.2 The Alert Start Time settings

The Alert Start Time settings determine the starting time for an alert. The Alert Start Time section includes six drop-down list boxes for setting the hour, minute, second, month, day, and year. The default settings are based on the system clock of the server where Wireless Edition is running. For example, this alert could notify whenever the price of a stock changed by 1.0, or whenever the price crossed a particular threshold, up or down.

Figure 6-7 The Create Alert Screen

Service > Alert > AlertAddress > LocationMark > Profile

Personalization > Service

Copy Service
Create Alert

Create Alert

* Name

Alert Address

Alert Sent Every

Alert Sent Every

Alert Starts At

Hour	Minute	Second
<input type="text" value="12"/>	<input type="text" value="6"/>	<input type="text" value="16"/>
Month	Day	Year
<input type="text" value="1"/>	<input type="text" value="9"/>	<input type="text" value="2001"/>

6.8.2 Editing Alert Parameters

You edit the parameters of an alert using the AlertList page. To edit the alert parameters:

1. Click the Alert tab to display the AlertList page. The AlertList displays previously created alerts in the Alert Address List.
2. From the Alert Address List, select an alert using the radio button.
3. Click the Edit button.
 - a. Rename the alert by entering a new name in the Name field.
 - b. Change the alert address setting using the drop-down list.
 - c. Adjust the alert frequency.
4. Click Apply to commit your changes and return to the AlertList page. Clicking the Cancel button suspends and returns to the AlertList page. Clicking the Delete button deletes the alert and returns to the AlertList page.

6.9 Managing Alert Addresses

Clicking on the AlertAddress tab displays a page which lists all previously created alerts and a Create button for assigning new Alert Addresses. The list contains:

- A Select, radio button column.
- An AddressName text box.
- An Address text box. Enter an email address if the alert should be sent to an email account. If the alert is to be sent to an SMS service, such as a WAP enabled phone, the service provider supplies the address (for example, 1-555-555-5555@phone.company.com).
- A Device drop-down list box, listing the alert delivery devices currently supported by Wireless Edition.
- A Default state column.

The AlertAddress list has three buttons which control its function, the Delete, SetDefault, and Apply buttons.

- The Delete button removes the AlertAddress.
- The SetDefault button makes this alert the default setting.

Figure 6–8 The Alert Address Page

Service ▾ Alert ▾ **AlertAddress** ▾ LocationMark ▾ Profile

Personalization > AlertAddress

Alert Address

Alert Address List

For Selected Alert Address Delete SetDefault Apply

Select	Address Name	Address	Device	Default
<input checked="" type="radio"/>	<input type="text" value="MyEmail"/>	<input type="text" value="MyEmail@MyEmailServi"/>	EMAIL ▾	true

Create a New Alert Address

* Address Name

* Address

Device ▾

Default

Create

6.9.1 Creating a New Alert Address

1. Click the Alert Address tab.
2. In the Create a New Alert Address section of the page, enter a name for the alert address in the Address Name field.

Note: You must have an address to which Wireless Edition can send content. This information is provided by the service vendor. For example, a service (SMS) address might be 1-555-555-5555@phone.company.com.

3. Enter the address for the new alert in the Address field.

Note: The Address Name and Address fields are required.

4. Select a device type from the Device drop-down list. The Device drop-down list displays a list of alert delivery logical devices that Wireless Edition supports for email and SMS.
5. Select the Default check box to make this Alert Address the default.
6. Click the Create button to create the AlertAddress. The new alert address then displays in the Alert Address List.

6.10 Creating a LocationMark

The form factor of some mobile devices limits the entry and display of such spatial information as street address and location coordinates. To solve this problem, the Wireless Edition stores spatial information as a locationmark, a name of a location that is meaningful to the user. For example, the locationmark *My Home* identifies "123 Main Street, Somewhere City, CA 12345.

Note: Location marks must be identified as a point for this release of Oracle9i Application Server Wireless Edition.

Locationmarks enable users to avoid entering lengthy alphanumeric strings on their mobile devices. Instead, users enter and manage the underlying spatial information for the locationmarks using the Personalization Portal. You can access this information by selecting the locationmark on your mobile device. The locationmarks are stored in the Wireless Edition repository.

Note: The locationmark feature does not function unless the server has access to geocoded data from a vendor.

To create a locationmark:

1. Click a service. The Edit Service page appears.

2. Click the LocationMark tab to invoke the LocationMark page.
3. In the LocationMark page, Click the Create button. The Create LocationMark page appears.

Figure 6–9 The Create LocationMark Page

Create LocationMark

* LocationMark Name	<input type="text" value="Oracle"/>
Description	<input type="text" value="Oracle Corp. Headquarters"/>
Label	<input type="text"/>
Company Name	<input type="text" value="Oracle Corp."/>
Address Line 1	<input type="text" value="500 Oracle Parkway"/>
Address Line 2	<input type="text"/>
City	<input type="text" value="Redwood City"/>
County	<input type="text" value="San Mateo"/>
State	<input type="text" value="CA"/>
Postal Code	<input type="text" value="94065"/>
Postal Code Ext	<input type="text"/>
Country	<input type="text" value="United States"/>
Default	<input checked="" type="checkbox"/>

4. Enter the following information into the LocationMark page:
 - a. Enter a name that is meaningful to in the LocationMark Name field. This is a required field.
 - b. Enter a description for the locationmark that is meaningful to you (for example, *Work* or *Home*).
 - c. Enter the name of the company in the Company Name field.
 - d. Enter the street address information in Address Line 1 and Address Line 2.

- e. Enter the county and state information (if applicable) in the County and State fields.
 - f. Enter the postal code in the Postal Code field. This can be a five-digit U.S. zip code or other postal code.
 - g. Enter the postal code extension (if applicable) in the Postal Code Ext. field.
 - h. Enter the country name in the Country name field.
5. Click the Create button to create the locationmark. The Wireless Edition checks the address data against the vendor's data and creates the locationmark. The new locationmark appears in the LocationMark List page. The LocationMark List contains the following elements:

Table 6–4 Elements in the LocationMark List

Element	List
Select	Selecting this radio button enables you to edit the locationmark with the Edit button, make the locationmark the default locationmark using the SetDefault button, or delete the locationmark using the Delete button.
LocationMarkName	This is the user supplied name for the locationmark.
LocationMarkDescription	The user supplied description, for example, <i>Work</i> or <i>Home</i> .
Default	Displays the status of the locationmark. If this is the default locationmark, the status is "true".

The LocationMark List also includes three buttons are activated by selecting a locationmark: Delete, SetDefault, and Edit. Clicking Delete deletes the selected locationmark. Clicking SetDefault sets the selected locationmark as the default locationmark. Clicking edit enables you to edit the properties of the locationmark.

Figure 6–10 The LocationMark List Page

Service > Alert > AlertAddress > **LocationMark** > Profile

[Personalization](#) > LocationMark

LocationMark List

For Selected LocationMark Delete SetDefault Edit

Select	LocationMark Name	LocationMark Description	Default
<input checked="" type="radio"/>	Oracle	Oracle Corp.	true

Create

6.11 Editing a LocationMark

To edit a locationmark:

1. From the LocationMark List, select the desired locationmark and then click the Edit button. The Edit LocationMark page displays. This page contains the same fields as Create Location Mark page, complete with the data used to create the selected locationmark. For more information, see [Section 6.10](#).
2. Edit the desired fields by clicking within a field to insert the cursor.
3. Click the Apply button to save the changes to the locationmark and return to the Location Mark List page. Clicking the Cancel button suspends all the changes and returns you to the LocationMark List page. Clicking on the Delete button deletes the locationmark.

6.12 Changing the Default Status of a LocationMark

To change the default status of a locationmark, choose a locationmark using the select column radio button, and click the SetDefault button to make it the default LocationMark. The status in the Default column of the LocationMark List changes to true. Only one LocationMark can be the default.

6.13 Deleting a LocationMark

To delete a locationmark, choose a locationmark using the select column radio button, and click the Delete button.

6.14 Editing a User Profile

Users can edit their own profiles by clicking the Profile page in the Personalization tab.

Figure 6–11 The User Profile Page

The screenshot shows a navigation bar with tabs: Service, Alert, AlertAddress, LocationMark, and Profile. Below the navigation bar is a breadcrumb trail: Personalization > Profile. The main heading is "User Profile". The form contains the following fields:

- * User Name: Text input with value "MySample"
- Display Name: Text input with value "Sample User"
- Old Password: Text input (empty)
- New Password: Text input (empty)
- Password Confirmation: Text input (empty)
- Password Hint: Text input (empty)
- Language: Dropdown menu with value "English"
- Country: Dropdown menu with value "United States"
- Default Alert Address: Dropdown menu with value "MyEmail"
- Default LocationMark: Dropdown menu with value "Oracle"
- Use Auto Location: Check box (unchecked)

At the bottom of the form is an "Apply" button.

To edit a user profile:

1. Click the Profile page in the Personalization tab. The User Profile page appears and displays the configuration information of the current user. The User Profile page contains the following parameters:

Table 6–5 Parameters for the User Profile Page

Parameter	Description
User Name	The name of the user. Note: this is case-sensitive
Display Name	The display name of the user.

Parameter	Description
Old Password	The user's previous password. Note: the password is case-sensitive.
New Password	The user's password. Note: the password is case-sensitive.
Password Confirmation	The user's password.
Password Hint	A prompt for the user's password.
Group	A list box entry appears for each group which can include this user.
Language	A drop-down list displaying the supported national languages.
Country	A drop-down displaying the options for user's country.
Default Alert Address	The address that alerts are sent to.
Default LocationMark	A drop-down list that displays the options for a default landmark.
Use Auto Location	This check box enables the user's location to be reported to the server for personalized content from a location-dependent service.

2. Change the user configuration parameters as needed:
 - a. Enter a new user name in the User Name field.
 - b. Enter a new display name for the user in the Display Name field.
 - c. Enter a new password for the user in the Password field.
 - d. Enter the new password again in the Password Confirmation field.
 - e. Enter a password hint that is meaningful to the user in the Password Hint field.
 - f. From the Group list box, select a new group for the user.
 - g. Select a language from the Language list box.
 - h. Select a country from the Country list box.
 - i. Select a default alert address from the Default Alert Address list box.

- j. Select a default locationmark address from the Default LocationMark list box.
 - k. Select the Auto Location check box to report the user's location to the server for personalized content from a location-dependent service. This is an optional step.
- 3. Click Apply.

Personalization from a Device

This document describes personalization from a device interface for customizing and assigning Wireless Edition mobile services. Each section of this document presents a different topic. These sections include:

- [Section 7.1, "Overview"](#)
- [Section 7.2, "Device Personalization"](#)
- [Section 7.3, "Logging in to the Portal"](#)
- [Section 7.4, "Creating a Folder"](#)
- [Section 7.5, "Creating a Bookmark"](#)
- [Section 7.6, "Reordering Objects"](#)
- [Section 7.7, "Renaming a Folder"](#)
- [Section 7.8, "Showing or Hiding Objects"](#)
- [Section 7.9, "Deleting a Folder"](#)
- [Section 7.10, "Editing a User Profile"](#)
- [Section 7.11, "Creating a LocationMark"](#)
- [Section 7.12, "Logging Off"](#)

7.1 Overview

The Wireless Edition portal can be personalized from a device, allowing a user to create the objects that Wireless Edition uses to perform its function.

All features of the Personalization Portal operate on a folder model. For example:

- A group is a folder containing users and services.

- A user is a folder containing sub-folders, services, location marks, bookmarks, and alerts.
- A service is an alias for a master service created using Service Designer.

User can create such objects as users, alerts, and locationmarks. In addition, the Personalization Portal enables users to perform service assignment, subfolder creation, and URL bookmark association, and manipulation of other Wireless Edition repository objects.

7.2 Device Personalization

The device-based Personalization Portal provides a subset of the current browser-based functionality to perform some user administration functions from a wireless device. For more information on the browser-based functionality of the Personalization Portal, see [Chapter 5, "Managing Users and Groups"](#) and [Chapter 6, "Personalization from Desktops"](#).

Note: The user's group must have access to the Personalization and UserAdmin adapters, or their equivalent, in order to utilize the Personalization Portal capabilities.

Figure 7–1 Home Screen for Device User

```
MySample
1▶MySampleFolder
2 MySampleFolder2
3 Personalization
4 User Admin

OK MyHome
```

7.2.1 Navigation and Data Entry

You use the device's up- and down-arrow keys to scroll through the lists. The current selection is indicated with the '>' symbol, or by using the number key which appears next to the item you wish to select. The key used to make the selection active varies from device to device. This document refers to the OK key for this function. It may be a single key used to accept selections, or one of two keys which correspond to an "OK" and "MyHome" text displayed in the corners of your device's screen. Pressing the button which appears under MyHome returns you to

the root level of the portal menu. Pressing the OK button accepts the selection. Pressing the MyHome button cancels any operation and returns to the top level display.

In order to begin using personalization options, you must select the Personalization folder. The Personalization folder houses most of the personalization options.

7.3 Logging in to the Portal

Logging in to the portal starts from the login screen. Your administrator provides you with a phone number or URL for this purpose. After you have connected, you must perform the following tasks:

1. Enter the user name and then press the OK key.
2. Enter the password and then press the OK key.
3. If you have entered something incorrectly, you are returned to that step to correct the error, otherwise you are logged in.

When you log in, your device displays the services enabled for your use. These may be services enabled by you using a browser to access the portal, or these may be services that your administrator has assigned to your group. Personalization is the name of the service that allows you to personalize the portal from a device.

The main level of the device portal is called "MyHome". You start at this level to perform any personalization or use any service. The MyHome folder can be restored at any time without harming any of the existing Personalization Portal settings.

7.4 Creating a Folder

Folders can store objects such as services, bookmarks, location marks, and services. To create a folder:

1. Select the [.]Home option.
2. Select the Add Folder option with the arrow keys, and then press the OK key to activate it.

Figure 7-2 Adding a Folder

```
MySample
1▶Add Folder
2 Add Bookmark
3 Reorder Services

-----
OK                MyHome
```

3. Using the device text entry keys, enter the name of the folder and press the OK key to submit the folder to the portal.
4. Wireless Edition creates the folder and returns to the [.]Home screen for the folder you started from.
5. Select the new folder and press OK. Notice that this folder is called [.]Folder_Name with Folder_Name being the name you gave your folder. Select the [.]Folder_Name option and press OK. Notice that you have many of the same options within this branch as in the root folder.

7.5 Creating a Bookmark

A bookmark is a link to a URL. To create a bookmark:

1. Select the [.]Home option.
2. Select the Add Bookmark option with the arrow keys, and press the OK key to activate it.
3. You are prompted to name the bookmark, then press OK.
4. The next screen prompts you for a URL.
5. Press OK to submit the bookmark to the portal.
6. Wireless Edition creates the bookmark and returns to the [.]Home screen for the folder you started from.

7.6 Reordering Objects

Once you have more than one service or folder, you may wish to rearrange the order in which they appear on the screen. To reorder objects:

Figure 7-3 Reordering Objects

```
Reorder Services
1▶MyHelloWorld
2 MySampleFolder
3 MySampleFolder2

OK MyHome
```

1. Select the [.]Home option at the folder level where the folders and services you are rearranging reside.
2. Scroll down and select the reorder option. The folders and services are displayed as they are currently organized. Items in this list are reordered upwards.
3. Scroll to the item you wish to move to select it and press OK. The list is rearranged with your selection moved up one level.

7.7 Renaming a Folder

To rename a folder:

1. Select the [.]Folder_Name option in the folder to be renamed. You are prompted to enter the new name.
2. Press OK to submit the new name to the portal.
3. Wireless Edition renames the folder and returns to the [.]Folder_Name screen for the folder from which you started.

7.8 Showing or Hiding Objects

When an object is first created it is hidden from view in the top-level screen, but is visible in the Personalization screen. Hidden folders are indicated by an asterisk (*). For example, in Figure 7-4, *MyHelloWorld and *Oracle are both hidden.

Figure 7–4 Hidden Objects

```

Edit
1▶[.]Home
2 *MyHelloWorld
3 [MySampleFolder]
4 [MySampleFolder2]
5 *Oracle
-----
OK                               MyHome
    
```

To hide or show an object:

1. Select the [.]Folder_Name option for the folder you wish to Hide or Show.
2. The list displays the state to be selected, for example, if the list says 'Hide' the folder is currently displayed but will become hidden if that option is selected.

Figure 7–5 Hide / Show State

```

*MyHelloWorld
1 Create Service
2 Create Alert
3 Rename
4▶Show
5 Delete
-----
OK                               MyHome
    
```

3. Scroll to the Show/Hide line, or press the appropriate number key. The folders state is changed, and is displayed, or not, when you return to the MyHome level.

7.9 Deleting a Folder

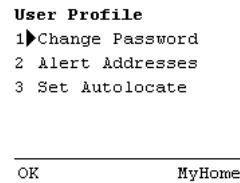
To delete a folder:

1. Select the [.]Folder_Name option in the folder to be removed.
2. Select the Delete option and press OK.
3. A warning appears before you can accept the deletion. Once deleted, that folder and all of its contents are erased from the Personalization Portal.

7.10 Editing a User Profile

At the Personalization folder level, there are two other options. The first of these is the User Profile option. Selecting this option displays two other options.

Figure 7-6 The User Profile Screen



- Change Password allows you to change the user password from your device.
- Alert Addresses allows you to create an Alert Address.
- Set Autolocate makes Wireless Edition's autolocate active or inactive.

7.10.1 Change Password

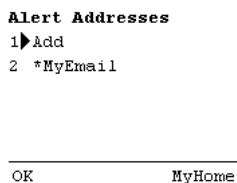
To change a password:

1. Select the Change Password option and press the OK key.
2. You are prompted to enter the Old Password.
3. You are prompted to enter the New Password.
4. You are prompted to Confirm Password.
5. If the Old Password does not match, then the process must be restarted from MyHome. If the Old Password matches, the password is changed and you are returned to the User Profile screen.

7.10.2 Assigning an Alert Address

An alert address defines how and where alerts are sent. Alerts can be sent as email format or short message service (SMS) format. While it is possible to add alert addresses from a device, it is much simpler to do so from the Personalization Portal Web browser interface. For more information on creating an alert using a browser interface, see [Section 6.9, "Managing Alert Addresses" in Chapter 6, "Personalization from Desktops"](#).

Figure 7–7 Creating an Alert Address



To create an alert address:

1. Select the Alert Address option and press OK.
2. You are prompted for the name that identifies the Alert Address.
3. You are prompted for the URL for the address.
4. Press the OK key to submit it to Wireless Edition.

7.10.3 Activating Autolocate

This option either activates or deactivates the Wireless Edition's autolocate feature. This feature queries your device for its current real location to use when providing location-based service information.

To manage the autolocation feature:

1. Autolocation is off by default, so the option visible in the display reads Autolocate On.
2. Select the Autolocate On to make active.
3. If Autolocate is active, the display selection to deactivate it reads Autolocate Off.

7.11 Creating a LocationMark

Locationmarks are a means of creating an alias for a location useful to the user. These can be added from a device using the LocationMark option. While it is possible to add locationmarks from a device, it is much simpler to do so from the Personalization Portal Web interface. For more information on creating a locationmark from the Web browser interface, see [Section 6.10, "Creating a LocationMark"](#) in [Chapter 6, "Personalization from Desktops"](#).

To create a Locationmark:

1. At the Personalization folder level, select the LocationMark option.
2. Select the Add option. All entries below Add are existing locationmarks.

Figure 7–8 Adding a LocationMark

```
Location Mark
1▶Add
2 Home
3 *Oracle

-----
OK           MyHome
```

3. You are prompted first for a Name. This name becomes the alias for the geographical data for this location.
4. You are prompted on sequentially displayed screens, for up to two address lines, city, county, state, and country information.
5. The locationmark is submitted to the portal. If it is accepted, you return to the locationmark screen, otherwise an error message is displayed.

7.12 Logging Off

The fastest way to log out of the Personalization Portal is to terminate the call or browser session.

Logical Devices

This document explains how to create and manage logical devices. Each section of this document presents a different topic. These sections include:

- [Section 8.1, "Overview"](#)
- [Section 8.2, "Creating a Logical Device"](#)
- [Section 8.3, "Modifying a Logical Device"](#)
- [Section 8.4, "Deleting a Logical Device"](#)
- [Section 8.5, "Determining the Client Device Type"](#)

8.1 Overview

A logical device is a repository object that represents either a physical device, such as a Nokia mobile phone, or an abstract device, such as email. Logical devices represent the interface between Wireless Edition transformers and the target devices or applications.

Wireless Edition determines the type of device that is requesting a service using the user agent parameter of the HTTP request. When adding support for a new device type, you generally need to:

1. Create the logical device in the repository.
2. Create a transformer for the device.
3. Configure Wireless Edition to recognize the new device.

In many cases, when adding support for a new device type, you can simply modify or re-use an existing transformer or logical device.

8.2 Creating a Logical Device

You use the Logical Devices form sequence in the Wireless Edition Service Designer to create a logical device in the repository.

To display the form sequence:

1. Right-click Logical Devices in the Wireless Edition repository tree view.
2. Click Create New Device to display the first form in the sequence.

Figure 8–1 The Logical Device Name and Display Data Form

The first form in the Create New Device sequence has the following parameters:

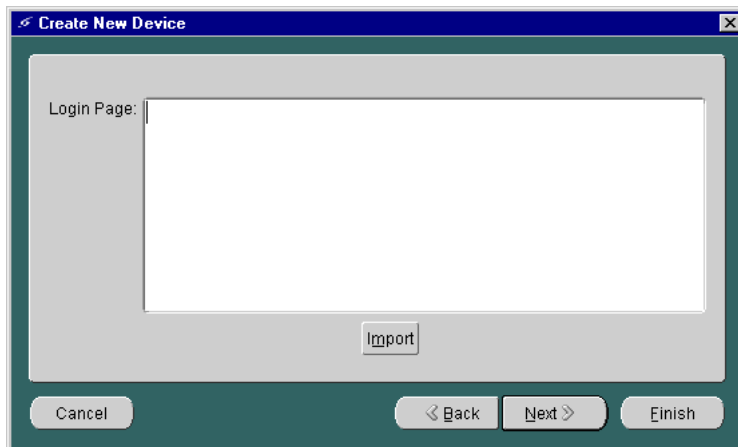
Table 8–1 Parameters for the Create New Device Screen

Parameter	Value
Name	The logical device name. This must be a unique name.
MIME Type	The logical device MIME type.
Encoding	Content encoding for the logical device. Content encoding is used to transport the result of the device type.

Parameter	Value
Needs URL Caching	When selected, the URL-encoding request parameters are cached on the server and only small pointers are encoded within the URL that is sent to the logical device. Use this option for devices with limited memory. You should also select this option if the browser does not support cookies.
Screen Width	Use this option to describe the width of the screen of the physical device in points.
Screen Height	Use this option to describe the height of the screen of the physical device in points.
Screen Columns	The screen size in characters. Use this option if you want to break text into pages.
Screen Rows	The screen size in characters. Use this option if you want to break text into pages.
Number of Softkeys	Allows users to map a hot link to a phone soft key.
Default Transformer	Specifies the transformer to use to convert content for the device.
Prolog	Specifies the format of the prolog required by the device. The prolog frequently includes processing instructions and meta tags.

The second form in the Create New Device sequence appears as follows:

Figure 8–2 The Login Page Form

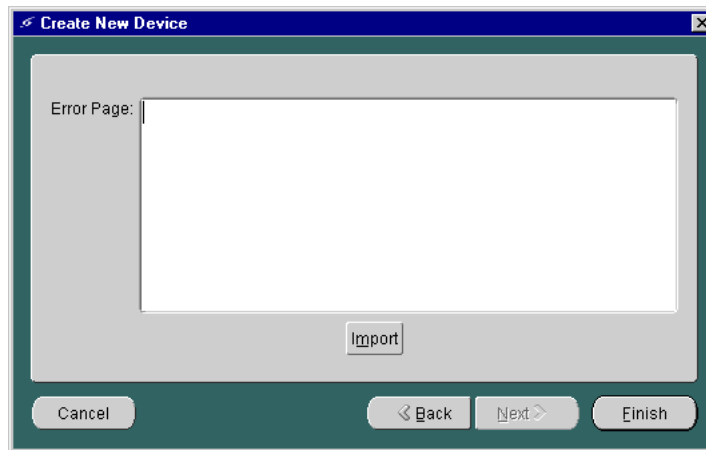


The second form in the Create New Device sequence includes the following parameter:

Table 8–2 Parameter of the Login Page

Parameter	Value
Login Page	The XSLT stylesheet that generates the login page for the device. Click Import and choose an XSLT stylesheet from the Open dialog.

The third form in the Create New Device sequence appears as follows:

Figure 8–3 The Error Page Form

The third form in the Create New Device sequence has the following parameter:

Table 8–3 Parameter of the Error Page

Parameter	Value
Error Page	The XSLT stylesheet that generates the error page for the device. Click Import and choose an XSLT stylesheet from the Open dialog.

Click Finish to create the object in the repository.

8.3 Modifying a Logical Device

The Wireless Edition Service Designer enables you to modify a logical device in the repository. To modify a logical device:

1. Expand the Logical Device folder in the Wireless Edition repository tree view.
2. Click the device you want to modify. The Edit Logical Device form appears in the right frame.
3. Modify the desired parameter(s). You can toggle between the General, Login Page, and Error Page tabs.
4. Click Apply.

8.4 Deleting a Logical Device

To delete a logical device from the repository:

1. Expand the Logical Device folder in the Wireless Edition repository tree view.
2. Right-click the device you want to delete.
3. Click Delete.
4. Click Yes in the Confirm Delete dialog to confirm the action.

8.5 Determining the Client Device Type

When Wireless Edition receives a service request, it determines the device type that submitted the request by querying the user agent parameter of the HTTP header. The **System.properties** file specifies the Wireless Edition class that queries the header and makes the determination:

```
locator.useragent.class=oracle.panama.core.xform.UserAgentImpl
```

The `UserAgentImpl` class is a generic implementation of the `UserAgent` interface. The `UserAgent` interface takes the HTTP user agent parameter and returns the device type as a string corresponding to the device name in the repository. If the device type is unknown, it returns NULL.

You can create your own implementation of the `UserAgent` interface to implement specialized logic for determining the client device type. For example, if the HTTP gateway that passes requests to Wireless Edition includes the telephone number of the client, you can implement a user agent that uses the telephone number to retrieve end user information from a directory.

Once you implement your own user agent, you must point the user agent class locator in the **System.properties** file to your implementation:

```
locator.useragent.class=oracle.panama.core.xform.MyUserAgentImpl
```

Wireless Edition uses a properties file, **useragent.properties**, to map the HTTP user agent parameter to a logical device in the Wireless Edition repository. The `UserAgentImpl` class retrieves this information from **useragent.properties** dynamically. When you add logical devices to the repository, you should map the device to the corresponding HTTP user agent parameter in the **useragent.properties** file.

To determine what information Wireless Edition actually receives in the HTTP header, you can check the Wireless Edition system log files. For full request

manager logging, set the request manager debug flag. Using the probe monitor, click Debug, then click “Change debug flag”.

Note: The **useragent.properties** file also specifies the default logical device that Wireless Edition uses if it cannot determine the device type. By default, this is set to WML11.

The **useragent.properties** file uses the following syntax to map an HTTP user agent parameter to a Wireless Edition logical device:

```
[ID].useragent=[UserAgent startstring from HTTP Request]
[ID].logicaldevice=[which logical device to use]
```

The logical device should appear on the line immediately following the corresponding user agent string. They should use the same ID, which is simply a unique number in the **useragent.properties** file. The ID associates a user agent string in the request with a logical device. For example:

```
30.useragent=MOTO
30.logicaldevice=Motorola
```

The `UserAgentImpl` class matches the actual header string to the longest matching user agent mapping in the **useragent.properties** file. That is, given the following mappings:

```
42.useragent=UP.Browser/3.01-IG01
42.logicaldevice=HDML

43.useragent=UP.Browser
43.logicaldevice=WML11
```

The `UserAgentImpl` class would match the header string “UP.Browser/3.01” to the HDML logical device, rather than the WML11 logical device.

Once you create or modify a device mapping, you can restart Wireless Edition or force a re-read of the property file using the probe runtime monitor (on port 8090), so that the changes to the property file take effect.

Walkthroughs

This document describes how to create Oracle9i Application Server Wireless Edition services. Each section of this document presents a different topic. These sections include:

- [Section 9.1, "Purpose"](#)
- [Section 9.2, "Before You Begin"](#)
- [Section 9.3, "Creating a SQL Service"](#)
- [Section 9.4, "Creating a PL/SQL Service"](#)
- [Section 9.5, "Creating a Master Service Using the URL Adapter"](#)
- [Section 9.6, "Creating Location-Based Services"](#)
- [Section 9.7, "Creating the Spatial Adapters"](#)
- [Section 9.8, "Creating a Master Service Using the Web Integration Adapter"](#)
- [Section 9.9, "Creating Chained Services Using the Web Integration Developer"](#)
- [Section 9.10, "Creating External Links Using the Service Designer"](#)

9.1 Purpose

The following walkthroughs provide step-by-step instructions for creating and distributing mobile services with the Wireless Edition. In these walkthroughs, you create services based on the Wireless Edition adapters. At the end of each walkthrough, you test your development using a device emulator.

These walkthroughs familiarize you with the Wireless Edition tools, including the Personalization Portal, the Master Service Creation Wizard, the Service Designer,

and the Web Integration Developer and how they enable you to create mobile services.

9.2 Before You Begin

Before you begin the walkthroughs, make sure that you correctly install and configure the Wireless Edition server. Also, you must complete the client installation on your development environment.

Start the Service Designer

To start the Wireless Edition Service Designer, from the Start menu, point to the Oracle for Windows NT menu item. Click Wireless Edition, then Service Designer.

Connect to the Wireless Edition Repository

When the Service Designer starts, the Log In dialog appears. Use the Log In dialog to connect the Service Designer to the Wireless Edition repository that you want to develop. The Log In dialog enables you to select a server name that you have already used, or to enter a new server name.

To connect to the Wireless Edition repository from the Log In dialog:

1. In the location field, enter the name of the Wireless Edition server. Then enter Administrator as the user name and manager as the password.

Note: Both user name and password are case-sensitive.

2. Click OK. The Wireless Edition server objects appear in the tree view on the left.

Note: The Service Designer displays an Error dialog for invalid entries. By clicking the Show Details button, you can view the Java runtime exception details.

Each section of the walkthrough describes a task needed to complete the creation of a service. These sections include:

9.2.1 Creating a User Group and a User

In this step, you create a user group and then a user. Using the Service Designer, you create an overall folder which will later contain a unique "root folder" for each user that you create. Each "root folder" will have the same name that you used to name the user. You can call the overall folder anything you like that is meaningful to you, keeping in mind that this folder will contain the root folders for all the users. In our example, the overall folder is called "Users Home." Create this folder in the Service Trees node of the repository tree in the left frame.

The root folder of a particular user contains the services (including aliases, master services, and folders) which the specified user can access.

One benefit is that if you have many users, you can create an overall folder for each grouping of users. For example, using the label "Users Home" as an illustration, you can create "Users Home A" for all users whose name starts with A, "Users Home B" for users whose name starts with B, and so on. In this fashion, grouping the users' root folders within higher-level folders minimizes the number of objects that are displayed in the Service Designer's tree view. (For performance reasons, the default view in the Service Designer's tree view does not display more than 100 objects of any type.)

Second, you use the Personalization Portal to create a both user group and a user

Create the Overall Folder for the Users' Root Folders

To create the overall folder for the users' root folders (using "Users Home" as an example):

1. Start the Service Designer.
2. Right-click Service Trees in the repository tree view.
3. Click Create New Folder to display the *Create New Folder* form.
4. Complete this form as follows:
 - a. In the Name field, enter Users Home.
 - b. Select the Valid check box.
 - c. Select the Visible check box.
5. Click Finish.

The folder named "Users Home" now appears under Service Trees in the repository tree.

Create the User Group

To create a user group:

1. Access the Personalization Portal using a Web browser.

Note: The default Personalization Portal URL is `http://9iASWEServer.domain/portal/Login.jsp`.

2. Enter Administrator as the user name, then enter manager as the password. Both the user name and password are case-sensitive. The Administration tab appears.
3. Select the Groups page.
4. In the Create a Group portion of the Groups page, enter MySampleGroup in the New Group Name field.
5. Click Create. The new group appears in the Group Name section.

Create the User

To create the user:

1. From the Administrator tab in the Personalization Portal, select the New User page.
2. Configure the user as follows:
 - a. In the Name field, type 1234.
 - b. For the Display Name, type 1234.
 - c. For the Password, type 1234.
 - d. For the Password Hint, type 1234.
 - e. Click the down arrow on the list box for the list of groups and select MySampleGroup, which you had created earlier.
 - f. Select English from the list box in the Language field.
 - g. Select United States from the list box in the Country field.
 - h. Select the Enabled check box.
 - i. Click create to complete the creation of the user. The User page of the Administration tab reappears.

3. Confirm that you have created a user by entering 1234 in the Find User field and click Search.

Once you have created both the group and the user, you are ready to create service using the Wireless Edition adapters.

9.3 Creating a SQL Service

In this walkthrough, you use the Master Service Creation Wizard to create a Wireless Edition service based on the SQL adapter. You create a service that checks the current time from the system database and displays it to the user.

Before you begin the walkthrough, make sure you can access a Wireless Edition server from your development environment running the Service Designer.

9.3.1 Task 1: Using the Master Service Creation Wizard

Start the Service Designer and follow these steps:

1. Click the Tools menu in the Service Designer and select Master Service Creation Wizard. The Introduction screen of the Master Service Creation Wizard appears which lists the necessary steps for the creation of a master service. Click Next to go to the *Selecting or Creating a Service Folder* screen.
2. Select the *Create a New Service Folder* radio button.
3. Click Next. The *Creating Folder* screen appears. Enter the following information in the In the New Folder Info section:
 - a. For the Name field, enter Database Services.
 - b. Select the Valid check box.
 - c. Select the Visible check box.
 - d. Enter a description of the service that is meaningful to you, such as System Time.
4. Click Next. The *Selecting or Creating a Service Folder* screen reappears. Click the Database Services folder and click Next. The Setting the Master Service Properties screen appears. Enter the following information in the Master Service Info section:
 - a. For the Name field, enter SQL Time.
 - b. Select the Valid check box.

- c. Select the Visible check box.
5. Click Next. The *Selecting an Adapter* screen appears.
6. Select SQLAdapter from the list of adapters and click Next. The screen for Setting the Init Parameters appears. Enter the following information into the fields.
 - a. For the JDBC_CONNECT_STRING, enter the JDBC connect string for the database on which to query, as follows:

```
jdbc:oracle:thin:@host_name:port:SID
```

Note: Be sure to insert all colons (for example, thin:@host).

- b. For the JDBC Driver, enter the Java DriverClass name (for example, Oracle thin driver, Oracle.jdbc.driver.oracle.driver)
- c. For the PASSWORD and USERNAME, type the password and user name of any database user. You can use the sample user 1234 with the password 1234.
- d. Under Type of Statement, enter Query.
- e. For the STATEMENT, type:


```
select sysdate from dual;
```
7. Click Apply. The *Setting the Input Parameters* screen appears. The Input parameters screen is populated with the input parameters for the SQL adapter. You do not need to modify any of the input parameters.
8. Click Next to go to the *Setting the Output Parameters* screen. The Output parameters screen is also populated with the output parameters for the SQL adapter. You do not need to modify any of the output parameters.
9. Click Finish. You have created a master service.

9.3.2 Task 3: Creating an Alias that Points to SQL Time

Using the Service Designer, you can make a master service accessible to a user by placing it in a service tree that a user owns or can access.

1. Right-click Users Home in the Service Designer repository tree.
2. Click Create New Alias to display the *Create New Alias* form.

3. Complete this form as follows:
 - a. In the Name field, enter My Database Services.
 - b. Click the Browse button next to the Service field. A Browse Services dialog appears. Expand the Master Services folder within the Browse Services dialog and select SQL TIME. Click OK.
 - c. Select the Valid check box.
 - d. Select the Visible check box.
4. Click Finish.
5. Click the Input Parameters tab in the right frame and select the User Customizable check box for SQL TIME.

This creates an alias object in the Wireless Edition repository. Aliases, like master services, have input parameters. You can use the input parameters of an alias to override any input values specified by the master service. For this walkthrough, you use the default settings provided by the master service.

9.3.3 Task 4: Making SQL Time Available to a Group

You use the Service Designer to make the service you created available to a group of users:

1. Expand the Groups folder in the Wireless Edition repository tree view.
2. Click MySampleGroup. The parameters for the group appear in the right frame.
3. Click the Service Folders tab in the right frame to display the Service Folders panel. The Service Folders panel includes two fields, Service Folders and Selected Service Folders.
4. Under Service Folders, expand the Services folder.
5. Click Database Services.
6. Click the right arrow (>) button to move SQL Time to the Selected Folders field.
7. Click Apply.
8. Click Yes in the Confirm Apply Changes dialog.

This makes the service available to all users in the group MySampleGroup. In [Section 9.3.4](#), you test this service as the user 1234.

9.3.4 Task 5: Testing SQL Time

In this walkthrough, you test the Wireless Edition service, SQL Time, using a browser and a phone simulator. Testing the service on a browser gives you a similar result to testing it on the phone simulator.

Testing the Service on a Browser

Log on to a browser to view the service you created. To access the browser:

1. From a Web browser, enter the following URL:

`http://9iASWEServer.domain/ptg/rm`

Note: The default listener port for the Oracle9i Application Server is 7777. The URL for this port is `http://9iASWEServer.domain:7777`. Refer to the Oracle9i Application Server documentation to change the HTTP listener to port 80.

2. In the browser, enter 1234 for the user name and 1234 for the user password. The services available to user 1234 appear in the left frame.
3. Click Database Services, then My Database Services.
4. Click SQL TIME.

The Wireless Edition retrieves and displays the current system time.

Testing the SQL Time on a Phone Simulator

You can use any phone simulator (for example, WML, HDML, CHTML) to test SQL Time. If you do not already have one, download and install a simulator. If you are working behind a firewall, you must configure the proxy server settings in the simulator before using it to access external sites.

Follow these steps to test the service from a phone simulator.

1. Start the simulator.
2. Enter the following URL in the Go window:

`http://9iASWEServer.domain/ptg/rm`

This is the URL of the device portal for your Wireless Edition installation.

Note: The default port is 7777. The URL for this port is `http://9iASWEServer.domain:7777/ptg/rm`.

3. Log in using 1234 as the user name and password. The preconfigured Wireless Edition services for user 1234 appear.
4. Select Database Services, then My Database Services.
5. Click SQL TIME.

The Wireless Edition retrieves and displays the current system time.

9.4 Creating a PL/SQL Service

Now that you have created a simple service, you can extend the functionality of the SQL adapter by creating a PL/SQL service. This walkthrough demonstrates how to pass parameters with PL/SQL using the Wireless Edition variables. Specifically, you use the sample PL/SQL package, **demo_proc.sql** (see [Section 9.4.1](#)), to create a service that allows you to enter a manager ID (the `input_manager_id` function in the **demo_proc.sql** package) that gets passed to the PL/SQL package's `employee_list` function using the Wireless Edition variables `p_manager_id`, `p_empno`, and `p_serv`.

After you have created this service, you then make it available to a group by creating an alias. Lastly, you test this service using the Personalization Portal and a phone simulator.

9.4.1 Task 1: Loading the PL/SQL Package, **demo_proc.sql**

Before you begin this walkthrough, you must load the following PL/SQL package, **demo_proc.sql** using such database tools as SQL*Plus. For example, you use the SQL*Plus command to load the package:

```
sqlplus demo/demo
SQL>@demo_proc
SQL>
```

The `demo_proc.sql` package is as follows:

```
CREATE OR REPLACE PROCEDURE demo_proc
  p_serv IN INTEGER          DEFAULT 0,
```

```

    p_empno IN INTEGER DEFAULT 0,
    p_manager_id IN INTEGER DEFAULT 0) AS
    v_first_name employee.first_name%TYPE;
    v_last_name employee.last_name%TYPE;
    v_manager_id employee.manager_id%TYPE;
    v_employee_id employee.employee_id%TYPE;

BEGIN
    dbms_output.enable(20000);
    IF (p_serv = 0)
        THEN GOTO input_manager_id;
    elsif (p_serv = 10)
        THEN GOTO employee_list;
    elsif (p_serv = 20)
        THEN GOTO options;
    elsif (p_serv = 30)
        then goto job_id;
    elsif (p_serv = 40)
        then goto manager_id;
    END IF;

<<input_manager_id>>
    dbms_output.put_line('<SimpleResult><SimpleContainer>');
    dbms_output.put_line('<SimpleForm target="__REQUEST_NAME__?PAoid=__PAoid__
    &#38;p_serv=10">');
    dbms_output.put_line('<SimpleFormItem name="p_manager_id" title="Enter ID"
    format="N*M">Enter ID:</SimpleFormItem>');
    dbms_output.put_line('</SimpleForm>');
    dbms_output.put_line('</SimpleContainer></SimpleResult>');
    GOTO fin;

<<employee_list>>
    select employee.first_name, employee.last_name INTO v_first_name, v_last_name
    FROM employee where employee.manager_id = p_manager_id;

    dbms_output.put_line('<SimpleResult><SimpleContainer>');
    dbms_output.put_line('<SimpleMenu title="EmpID First Last">');
    FOR i IN (select employee_id, first_name, last_name from employee where manager_
    id = p_manager_id)
    LOOP
        dbms_output.put_line('<SimpleMenuItem title="Employees" target="__REQUEST_
    NAME__?PAoid=__PAoid__&#38;p_serv=20&#38;p_empno='||to_char(i.employee_
    id)||'">Employee '||i.employee_id||' is '||v_first_name||' '||v_last_name||'
    </SimpleMenuItem>');
    END LOOP;

```

```

dbms_output.put_line('</SimpleMenu>');
dbms_output.put_line('</SimpleContainer></SimpleResult>');
GOTO fin;

<<options>>
  dbms_output.put_line('<SimpleResult><SimpleContainer>');
  dbms_output.put_line('<SimpleMenu title="Request ' || p_empno || '">');
  dbms_output.put_line('<SimpleMenuItem target="__REQUEST_NAME__?PAoid=__PAoid__&#38;p_serv=30&#38;p_empno=' || p_empno || '">Update Salary</SimpleMenuItem>');
  dbms_output.put_line('<SimpleMenuItem target="__REQUEST_NAME__?PAoid=__PAoid__&#38;p_serv=40&#38;p_empno=' || p_empno || '">Update Manager ID</SimpleMenuItem>');
  dbms_output.put_line('</SimpleMenu>');
  dbms_output.put_line('</SimpleContainer></SimpleResult>');
  GOTO fin;

<<job_id>>
select employee.last_name INTO v_last_name FROM employee where employee_id = p_
empno;

update employee set salary = 100 where employee_id = p_empno;

commit;

  dbms_output.put_line('<SimpleResult><SimpleContainer>');
  dbms_output.put_line('<SimpleText wrapit="true" title="Changed to 100">');
  dbms_output.put_line('<SimpleTextItem>Employee ' || p_empno || ' ' || v_last_name || '
salary changed </SimpleTextItem>');
  dbms_output.put_line('</SimpleText>');
  dbms_output.put_line('<SimpleHref type="options" title="Prev" target="__
REQUEST_NAME__?PAoid=__PAoid__"/>');
  dbms_output.put_line('</SimpleContainer></SimpleResult>');
  GOTO fin;

<<manager_id>>
  dbms_output.put_line('<SimpleResult><SimpleContainer>');
  dbms_output.put_line('<SimpleText wrapit="true" title="Changed to 7788">');
  dbms_output.put_line('<SimpleTextItem>Employee ' || p_empno || ' manager changed
</SimpleTextItem>');
  dbms_output.put_line('</SimpleText>');
  dbms_output.put_line('<SimpleHref type="options" title="Prev" target="__
REQUEST_NAME__?PAoid=__PAoid__"/>');
  dbms_output.put_line('</SimpleContainer></SimpleResult>');
  GOTO fin;

<<fin>>

```

```
    null;  
  
END demo_proc;  
/  
show errors;
```

9.4.1.1 Sample Code Explanation

The PL/SQL package, **demo_proc.sql**, shows how the Wireless Edition, using PL/SQL, passes variables to update a table.

Sample Scenario

Using a mobile phone, a manager accesses a list of employees and then updates the salary of an individual employee. To begin the process, the manager requests the service on the mobile phone. (The service is the name of the Wireless Edition alias created using the Service Designer. For more information, see [Section 9.4.3](#).)

The first screen of the service requests the manager's ID. After the manager enters this value, the service presents the manager with a screen that lists the employees that report to the manager.

The manager selects Employee List and the manager ID queries the database for all employees who report to that manager (for example, employees who have been assigned that manager ID). The service retrieves a list of employees. The manager then selects an individual employee from this list. A screen then appears, presenting the manager with a menu with options to update the employees's salary or the manager's ID. The manager then chooses the option to update the employee's salary. The manager decides to increase the selected employee's salary by \$100 and enters 100.

Defining PL/SQL Variables

In **demo_proc.sql**, Wireless Edition defines the user-input variables as `p_serv`, `p_empno`, and `p_manager` as follows.

```
CREATE OR REPLACE PROCEDURE demo_proc  
    p_serv      IN INTEGER          DEFAULT 0,  
    p_empno     IN INTEGER          DEFAULT 0,  
    p_manager_id IN INTEGER          DEFAULT 0)
```

These variables are defined using the Master Service Creation Wizard. For more information, see Step 3e under [Section 9.4.2](#). The variable `p_serv` defines the code branches in this PL/SQL package as follows:

```
IF (p_serv = 0)
```



```

        THEN GOTO input_manager_id;
    elsif (p_serv = 10)
        THEN GOTO employee_list;
    elsif (p_serv = 20)
        THEN GOTO options;
    elsif (p_serv = 30)
        then goto job_id;
    elsif (p_serv = 40)
        then goto manager_id;
    END IF;

```

The variable `p_serv` navigates the above menu list by invoking the URL appropriate to the input value. For example, when a user selects the employee list, `p_serv` is passed in as 10, which invokes the URL containing the list of employees through the following function, which defines the target URL.

```
target="__REQUEST_NAME__?PAoid=__PAoid_&#38;p_serv=10">'
```

Note: Three separate underscore marks (_ _ _) must both precede and follow the `REQUEST_NAME` function. The function is also case-sensitive.

The target URL function passes variables from one page or screen to another from one page or screen to the next using `&`, the equivalent to the "&" in the URL.

Defining the Output

The following string is an example of how to write the Wireless Edition XML document from PL*SQL.

```
dbms_output.put_line('<SimpleResult><SimpleContainer>');
```

For more information on the Wireless Edition DTD (`SimpleResult` DTD), see the *Oracle9i Application Server Wireless Edition Developer's Guide*.

9.4.2 Task 2: Creating the Master Service

Invoke the Master Service Creation Tool as you did in [Section 9.3.1, "Task 1: Using the Master Service Creation Wizard"](#).

1. From the *Select or Create a New Service Folder* screen (Step 1), select the Database Services folder you created in [Section 9.3.1](#). The *Setting the Master Service*

Properties screen appears. Enter the following information in the Master Service Info section:

- a. For the Name field, enter PLSQL Test.
 - b. Select the Valid check box.
 - c. Select the Visible check box.
2. Click Next. The *Selecting an Adapter* screen appears.
 3. Select SQL adapter from the list of adapters and click Next. The screen for Setting the Init Parameters appears. Enter the following information into the fields.

- a. For the JDBC_CONNECT_STRING, enter the JDBC connect string for the database on which to query, as follows:

```
jdbc:oracle:thin:@host_name:port:SID
```

Note: Be sure to insert all colons (for example, thin:@host).

- b. For the JDBC Driver, enter the Java DriverClass name (for example, Oracle thin driver, Oracle.jdbc.driver.oracle.driver)
- c. For the PASSWORD and USERNAME, type the password and user name of any database user. For example, You can use the sample user demo with the password demo.

Note: This is a database user.

- d. Under Type of Statement, select PL/SQL from the drop-down menu.
- e. For the STATEMENT, type:

```
BEGIN demo_proc(:p_serv, :p_empno, :p_manager_id);END;
```

Figure 9–1 The *Init Parameters* screen for the SQL Adapter

4. Click Apply. The *Setting the Input Parameters* screen appears. The Input parameters screen is populated with the input parameters for the SQL adapter.

Note: The Value field in the Input Parameters screen sets default parameters such as 0. For example, setting `p_serv` equal to 0 results in a service that starts with 0 until it is changed by users.

5. Click Next to go to the Setting the Output Parameters screen. Click Next.
6. Click Finish. You have created the master service, PLSQL.

9.4.3 Task 3: Creating an Alias that Points to PLSQL Test

Next you can make a master service accessible to a user by placing it in a service tree that a user owns or can access.

1. Right-click the folder, PLSQL Test in the Wireless Edition repository tree.
2. Click Create New Alias to display the *Create New Alias* form.

3. Complete this form as follows:
 - a. In the Name field, enter PLSQL Test Service.
 - b. Click the Browse button next to the Service field. A Browse Services dialog appears. Expand the Master Services folder within the Browse Services dialog and select PLSQL Test. Click OK.
 - c. Select the Valid check box.
 - d. Select the Visible check box.
4. Click Finish.

Click the Input Parameters tab in the right frame and select the User Customizable check box for PLSQL Test .

9.4.4 Task 4: Making PLSQL Test Available to a Group

To make PLSQL Test available to a group:

1. Expand the Groups folder in the Wireless Edition repository tree view.
2. Click MySampleGroup. The parameters for the group appear in the right frame.
3. Click the Service Folders tab in the right frame to display the Service Folders panel. The Service Folders panel includes two fields, Service Folders and Selected Service Folders.
4. Under Service Folders, expand the Services folder.
5. Click Database Services.
6. Click the right arrow (>) button to move PLSQL Test to the Selected Folders field.
7. Click Apply.
8. Click Yes in the Confirm Apply Changes dialog.

This makes the service available to all users in the group MySampleGroup.

9.4.5 Task 5: Testing PLSQL Test

In this walkthrough, you test the Wireless Edition service, PLSQL Test, using a browser and a phone simulator. Testing the service on a browser will produce the same results as testing it using a phone simulator.

Testing PLSQL Test on a Browser

Log on to the Personalization Portal to view PLSQL Test. To access the browser:

1. From a Web browser, enter the following URL:

`http://9iASWEServer.domain/ptg/rm`

Note: The default listener port for the Oracle9i Application Server is 7777. The URL for this port is `http://hostname.domain:7777`. Refer to the Oracle9i Application Server documentation to change the HTTP listener to port 80.

2. In the browser, enter 1234 for the user name and 1234 for the user password. The services available to user 1234 appear in the left frame.
3. Click Database Services, then My Database Services.
4. Click PLSQL Test.
5. Enter Manager ID.

Testing PLSQL Test on a Phone Simulator

You can use any phone simulator (for example, WML, HDML, CHTML) to test PLSQL Test. If you do not already have one, download and install a simulator. If you are working behind a firewall, you must configure the proxy server settings in the simulator before using it to access external sites.

Follow these steps to test the service from a phone simulator.

1. Start the simulator.
2. Enter the following URL in the Go window:

`http://9iASWEServer.domain/ptg/rm`

This is the URL of the device portal for your Wireless Edition installation.

Note: The default port is 7777. The URL for this port is `http://9iASWEServer.domain:7777/ptg/rm`.

3. Log in using 1234 as the user name and password. The preconfigured Wireless Edition services for user 1234 appear.
4. Select Database Services, then My Database Services.

Note: If the phone simulator returns an HTTP error, you should refresh the cookie cache and source cache. Go to the phone simulator install directory and refresh (or delete) the files **CookieCache** and **SourceCache**.

5. Click PLSQL Test.
6. Enter Manager ID.

9.5 Creating a Master Service Using the URL Adapter

In this walkthrough, you create a Wireless Edition service based on the URL adapter, which allows applications to invoke other URLs and pass in arguments. In addition the URL adapter enables the following functions:

- Integration with existing Java, JSP, or XML applications.
- Parameterization and cookie management, enabling remote applications to become wirelessly available.

The URL adapter is well suited for developers wishing to build a wireless Web site or internet application using only Java servlets and JavaServer Pages (JSPs).

This walkthrough uses a sample Java package that shows how to embed SimpleResult DTD. The JAVA program in the section is an example.

Note: The URL adapter requires input to be formatted as SimpleResult.

9.5.1 Task 1: Building the Java Program

Before you create the master service, you must build the following JAVA program.

```
import javax.servlet.*;
import javax.servlet.http.*;

import java.io.*;
```

```
import java.util.Enumeration;
import java.util.StringTokenizer;
import java.util.Properties;
import java.util.Date;

public class
CompanionServer
extends HttpServlet
{
    String directory = "";
    static String version = "v1.0";
    static String servletname = "/cmp/go";

    Properties properties;

    public void
    init (ServletConfig config)
    throws ServletException
    {
        super.init(config);
        return;
    }

    public void
    doGet (HttpServletRequest req,
          HttpServletResponse res)
    throws ServletException, IOException
    {
        String state = req.getParameter("state");

        PrintWriter o = res.getWriter();

        if (state == null)
        {
            res.setContentType("text/xml");
            print_default(req, res, o);
        }
        else if (state.equals("menu"))
        {
            res.setContentType("text/xml");
            print_menu(req, res, o);
        }
        else if (state.equals("form"))
        {
            res.setContentType("text/xml");
        }
    }
}
```

```

        print_form(req, res, o);
    }
    else if (state.equals("text"))
    {
        res.setContentType("text/xml");
        print_text(req, res, o);
    }
    else if (state.equals("test"))
    {
        res.setContentType("text/xml");
        print_test(req, res, o);
    }
    else if (state.equals("param"))
    {
        res.setContentType("text/xml");
        print_parameters(req, res, o);
    }
    else
    {
        res.setContentType("text/xml");
        // ERROR: unknown application state
        print_error(o, "unknown application state [get][\"+state+\"]");
    }
}

return;
}

void
print_default (HttpServletRequest req,
              HttpServletResponse res,
              PrintWriter o)
throws ServletException, IOException
{
    String server = req.getServerName();
    int port = req.getServerPort();

    print_header(o);
    o.println ("<SimpleText><SimpleTextItem>Click GO! to
Start</SimpleTextItem></SimpleText>");
    o.println ("<SimpleHref target=\"__REQUEST_NAME__?PAoid=__PAoid__
&#38;URL=http://\" + server + ":" + port + servletname + "?state=menu\"
title=\"GO!\" type=\"accept\"/>");
    print_footer(o);

    return;
}

```



```
}

void
print_parameters (HttpServletRequest req,
                 HttpServletResponse res,
                 PrintWriter o)
throws ServletException, IOException
{
    print_header(o);
    {
        Enumeration e = req.getParameterNames();

        while (e.hasMoreElements())
        {
            String name = (String) e.nextElement();
            String[] value = req.getParameterValues(name);

            for (int i = 0; i < value.length; i++)
            {
                o.println("<SimpleText>");
                o.println("<SimpleTextItem title=\"" + name + "\">");
                o.println(value[i]);
                o.println("</SimpleTextItem>");
                o.println("</SimpleText>");
            }
        }
    }
    print_footer(o);

    return;
}

void
print_header(PrintWriter o)
throws ServletException, IOException
{
    o.println("<SimpleResult><SimpleContainer>");
    return;
}

void
print_footer(PrintWriter o)
throws ServletException, IOException
{
    o.println("</SimpleContainer></SimpleResult>");
}
```

```

        return;
    }

    void
    print_error(PrintWriter o, String e)
    throws ServletException, IOException
    {
        print_header(o);
        o.println("<SimpleText><SimpleTextItem title=\"Error\">");
        o.println(e);
        o.println("</SimpleTextItem></SimpleText>");
        print_footer(o);

        return;
    }

    void
    print_menu(HttpServletRequest req,
                HttpServletResponse res,
                PrintWriter o)
    throws ServletException, IOException
    {
        String server = req.getServerName();
        int port = req.getServerPort();

        print_header(o);
        o.println("<SimpleMenu title=\"Menu\">");
        o.println("<SimpleMenuItem target=\"__REQUEST_NAME__?PAoid=__PAoid__
&#38;URL=http://\" + server + ":" + port + servletname +
"?state=text&#38;item=ONE\">");
        o.println("Choice 1");
        o.println("</SimpleMenuItem>");
        o.println("<SimpleMenuItem target=\"__REQUEST_NAME__?PAoid=__PAoid__
&#38;URL=http://\" + server + ":" + port + servletname +
"?state=text&#38;item=TWO\">");
        o.println("Choice 2");
        o.println("</SimpleMenuItem>");
        o.println("<SimpleMenuItem target=\"__REQUEST_NAME__?PAoid=__PAoid__
&#38;URL=http://\" + server + ":" + port + servletname +
"?state=text&#38;item=THREE\">");
        o.println("Choice 3");
        o.println("</SimpleMenuItem>");
        o.println("</SimpleMenu>");
    }

```

```

        print_footer(o);

        return;
    }

}

void
print_text (HttpServletRequest req,
            HttpServletResponse res,
            PrintWriter o)
throws ServletException, IOException
{
    String sitem = req.getParameter("item");
    String server = req.getServerName();
    int    port   = req.getServerPort();

    print_header(o);
    o.println("<SimpleText title=\"Result\">");
    o.println("<SimpleTextItem>" + sitem + "</SimpleTextItem>");
    o.println("</SimpleText>");
    o.println ("<SimpleHref target=\"__REQUEST_NAME__?PAoid=__PAoid__
    &#38;URL=http://" + server + ":" + port + servletname + "?state=menu\"
    title=\"Menu\" type=\"accept\"/>");
    print_footer(o);

    return;
}
}

```

9.5.1.1 Java Example Overview

The example presents URLs as menu items using Wireless Edition DTD, enabling users to select another URL or application. The target URL, defined by the following function, is the entry point to call the servlet.

```
target=\"__REQUEST_NAME__?PAoid=__PAoid__&#38;URL=http://" + server + ":" +
port + servletname + "?state=menu\">Menu</SimpleHref>"
```

The target URL function allows applications to invoke a URL. In addition, the target URL function passes arguments from one URL or application to another using the &, the equivalent to the "&" in the URL.

The `o.println` function defines the output as SimpleResult DTD as follows:

```
o.println ( <SimpleText><SimpleTextItem>Click GO! to
           start</SimpleTextItem></SimpleText> " );
```

9.5.1.2 Running the Java Example

To run the Java example with your web server:

1. Open the **jserv.conf** file
2. In the Mount section, add ApJServMount /cmp /companion
3. In the **jserv.properties** file, add companion to the servlet zones, root, stock, and companion.
4. Add the following reference to the companion zone configuration file:
`companion.properties=C:\Apache\JServ 1.1.1\servlets\companion.properties`

Note: Unlike normal Java properties, JServ configurations have some important extentions:

- Commas are used as token separators.
 - Multiple definitions of the same key are concatenated in a comma-separated list.
-
-

5. Add the following to the **companion.properties** file:

Table 9–1 Configuration Parameters

Parameter	Description
List of Repositories	<p>The list of servlet repositories controlled by this servlet zone. The default is NONE. The syntax is:</p> <pre>repositories=[repository],[repository]</pre> <p>For example:</p> <pre>repositories=C:\Apache\JServ 1.1.1\servlets,C:\PTG\companion\servlet</pre> <p>Place the classes to reload upon modification here.</p>

Parameter	Description
ClassLoader	<p>Enable servlet class autoreloading. The default is true. The syntax is:</p> <pre>autoreload.classes=[true,false] (boolean)</pre> <p>For example:</p> <pre>autoreload.classes=true</pre> <p>Enable servlet resourced autoreloading (properties and other loaded resources). The default is true. The syntax is:</p> <pre>autoreload.file=[true,false] (boolean)</pre> <p>For example:</p> <pre>autoreload.file=true</pre>
Initialization Timeout	<p>Set the number of milliseconds to wait before giving up on initializing a servlet. The default is 10000 (10 seconds). A time out of zero means no time out. The syntax is:</p> <pre>init.timeout=(long)>0</pre> <p>For example:</p> <pre>init.timeout=10000</pre> <p>Set the number of milliseconds to wait before giving up on destroying a servlet. The default is 10000 (10 seconds). The syntax is:</p> <pre>destroy.timeout=(long)>0</pre> <p>For example:</p> <pre>destroy.timeout=10000</pre>
Cookies	<p>Set whether or not to use cookies to maintain session state. If false, then <code>response.encodeUrl()</code> is the method to maintain session state. If true, then the servlet engine attempts to set a cookie when <code>request.getSession(true)</code> is called. The default is true. The syntax is:</p> <pre>session.useCookies=[true,false] (boolean)</pre> <p>For example:</p> <pre>session.useCookies=true</pre>
Session Timeout	<p>Set the number of milliseconds to wait before invalidating an unused session. The default is 1800000 (30 minutes). The syntax is:</p> <pre>session.timeout=(long)>0</pre> <p>For example:</p> <pre>session.timeout=1800000</pre>

Parameter	Description
Timed-Out Sessions	<p>Set how frequently (in milliseconds) to check for timed-out sessions. The default is 30000 (30 seconds). The syntax is:</p> <pre>session.checkFrequency=(long)>0</pre> <p>For example:</p> <pre>session.checkFrequency=30000</pre>
Domain Header	<p>Set the domain= header that gets sent with the cookie. This is optional. The default is null. The syntax is:</p> <pre>session.topleveldomain=.foo.com</pre>
SingleThreadModel Servlets Pool	<p>Set the initial capacity of the STM servlets pool. The default is 5. The syntax is:</p> <pre>singleThreadModelServlet.initialCapacity=(int)>1</pre> <p>For example:</p> <pre>singleThreadModelServlet.initialCapacity=5</pre>
Instances for Pool	<p>Set the number of servlet instances to be added to the pool if found empty. The default is 5. The syntax is:</p> <pre>singleThreadModelServlet.incrementCapacity=(int)>1</pre> <p>For example:</p> <pre>singleThreadModelServlet.incrementCapacity=5</pre>
Capacity of Pool	<p>Set the maximum capacity of the STM pool. The default is 10. The syntax is:</p> <pre>singleThreadModelServlet.maximumCapacity=(int)>1</pre> <p>For example:</p> <pre>singleThreadModelServlet.maximumCapacity=10</pre>

Table 9–2 Servlet Parameters

Parameter	Description
Startup Servlets	<p>Comma or space delimited list of servlets to launch on startup. This can either be a class name or alias. The default is NONE. The syntax is:</p> <pre>servlets.startup=[classname or alias],[classname or alias]</pre> <p>For example:</p> <pre>servlets.startup=hello,snoop,org.fool.Dummy</pre>
Servlet Aliases	<p>This defines aliases from which servlets can be invoked. Each alias gives a new instance of the servlet. This means that if a servlet is invoked both by class name and by alias name, it results in two instances of the servlet being created. The default is NONE. The syntax is:</p> <pre>servlet.[alias].code=[classname] (String)</pre> <p>For example:</p> <pre>servlet.snoop.code=SnoopServlet servlet.hello.code=org.fool.Dummy servlet.go.code=CompanionServer</pre>
Global Init Parameters	<p>Parameters passed here are given to each of servlets. You should put configuration information that is common to all servlets. The value of the property is a comma delimited list of "name=value" pairs that are accessible to the servlet via the method <code>getInitParameter()</code> in <code>ServletConfig</code>. The default is NONE. The syntax is:</p> <pre>servlets.default.initArgs=[name]=[value],[name]=[value],...</pre> <p>For example:</p> <pre>servlets.default.initArgs=common.to.everybody=Hi everybody!</pre>
Servlet Init Parameters	<p>These properties define init parameters for each servlet that is invoked by its classname. The default is NONE. The syntax is:</p> <pre>servlet.[classname].initArgs=[name]=[value],[name]=[value],...</pre> <p>For example:</p> <pre>servlet.org.fool.Dummy.initArgs=message=I'm a dummy servlet</pre>

Parameter	Description
Aliased Servlet Init Parameters	<p>These properties define init parameters for each servlet that is invoked by its alias. The default is NONE. The syntax is:</p> <pre>servlet.[alias].initArgs=[name]=[value],[name]=[value],...</pre> <p>For example:</p> <pre>servlet.snoop.initArgs=message=I'm a snoop servlet servlet.hello.initArgs=message=I say hello world to everyone servlet.go.initArgs=directory=C:\PTG\companion\ servlet</pre>

9.5.2 Task 2: Creating the Master Service

Invoke the Master Service Creation Tool as you did in [Section 9.3.1, "Task 1: Using the Master Service Creation Wizard"](#).

1. Select the *Create a New Service Folder* radio button in the *Selecting or Creating a Service Folder* screen.
2. Click Next. The Creating Folder screen appears. Enter the following information in the In the *New Folder Info* section:
 - a. For the Name field, enter URL Services.
 - b. Select the Valid check box.
 - c. Select the Visible check box.
 - d. Enter a description of the service that is meaningful to you.
3. Click Next. The *Selecting or Creating a Service Folder* screen reappears. Click the URL Service folder and click Next. The *Setting the Master Service Properties* screen appears. Enter the following information in the Master Service Info section:
 - a. For the Name field, enter URL Example.
 - b. Select the Valid check box.
 - c. Select the Visible check box.
4. Click Next. The *Selecting an Adapter* screen appears.
5. Select URLAdapter from the list of adapters and click Next.

6. Continue clicking the Next button until you complete the Master Service Creation Wizard Sequence.

Note: The URL of the XML document that is retrieved by the URL Adapter. The XML document should conform to the SimpleResult DTD.

For example, if the URL parameter is `file:///d:/SimpleResultExample1.xml`, then the adapter gets the file `d:/SimpleResultExample1.xml` from the hard drive on the local machine. If the URL parameter is `http://ptg1.oracle.com/SimpleResultExample2.xml`, then the adapter gets the file `SimpleResultExample2.xml` from the host `ptg1.oracle.com`.

9.5.3 Task 3: Creating an Alias that Points to URL Adapter

Next you can make a master service accessible to a user by placing it in a service tree that a user owns or can access.

1. Right-click the folder, URL Services in the Wireless Edition repository tree.
2. Click Create New Alias to display the *Create New Alias* form.
3. Complete this form as follows:
 - a. In the Name field, enter My URL Services.
 - b. Click the Browse button next to the Service field. A Browse Services dialog appears. Expand the Master Services folder within the Browse Services dialog and select URL Example. Click OK.
 - c. Select the Valid check box.
 - d. Select the Visible check box.
4. Click Finish.

9.5.4 Task 4: Making URL Example Available to a Group

To make URL Example available to a group:

1. Expand the Groups folder in the Wireless Edition repository tree view.
2. Click MySampleGroup. The parameters for the group appear in the right frame.
3. Click the Service Folders tab in the right frame to display the Service Folders panel. The Service Folders panel includes two fields, Service Folders and Selected Service Folders.

4. Under Service Folders, expand the Services folder.
5. Click URL Services.
6. Click the right arrow (>) button to move URL Services to the Selected Folders field.
7. Click Apply.
8. Click Yes in the Confirm Apply Changes dialog.

This makes the service available to all users in the group MySampleGroup.

9.5.5 Task 5: Testing URL Example

In this walkthrough, you test the Wireless Edition service, URL Example, using a browser and a phone simulator.

Testing the Service on a Browser

Log on to a browser to view URL Example. To access the service:

1. From a Web browser, enter the following URL:

```
http://9iASWEServer.domain/ptg/rm
```

Note: The default listener port for the Oracle9i Application Server is 7777. The URL for this port is `http://hostname.domain:7777`. Refer to the Oracle9i Application Server documentation to change the HTTP listener to port 80.

2. In the browser, enter 1234 for the user name and enter 1234 for the password. The services available to user 1234 appear in the left frame.
3. Click URL Services, then My URL Services.

Testing Geocoder on a Phone Simulator

You can use any phone simulator (for example, WML, HDML, CHTML) to test Geocoder. If you do not already have one, download and install a simulator. If you are working behind a firewall, you must configure the proxy server settings in the simulator before using it to access external sites.

Follow these steps to test the service from a phone simulator.

1. Start the simulator.
2. Enter the following URL in the Go window:
`http://9iASWEServer.domain/ptg/rm`
This is the URL of the device portal for your Wireless Edition installation.

Note: The default port is 7777. The URL for this port is
`http://9iASWEServer.domain:7777/ptg/rm`.

3. Log in using 1234 as the user name and password. The preconfigured Wireless Edition services for user 1234 appear.
4. Select URL Services, then My URL Services.

9.6 Creating Location-Based Services

In this walkthrough, you create a Wireless Edition service based on a spatial adapter, the Geocoder. In addition, you make this service location-dependent using the Region Modeling Tool.

9.6.1 Task 1: Creating the Location-Dependent Master Service

Invoke the Master Service Creation Tool as you did in [Section 9.3.1, "Task 1: Using the Master Service Creation Wizard"](#).

Create a New Folder

1. Select the *Create a New Service Folder* radio button in the *Selecting or Creating a Service Folder* screen.
2. Click Next. The Creating Folder screen appears.
3. Enter the following information in the *In the New Folder Info* section:
 - a. For the Name field, enter Spatial Services.
 - b. Select the Valid check box.
 - c. Select the Visible check box.
 - d. Enter a description of the service that is meaningful to you.
 - e. Click Next. The *Select or Create a New Service Folder* reappears.

Creating the Master Service

1. From the *Select or Create a New Service Folder* screen, select the Spatial Services folder you created in [Section 9.3.1](#). The Setting the Master Service Properties screen appears.
2. Enter the following information in the Master Service Info section:
 - a. For the Name field, enter Geocoder.
 - b. Select the Valid check box.
 - c. Select the Visible check box.
 - d. Enter a name for the Service that is meaningful to you.

Creating the User-Defined Region

1. Click the Location Dependent check box. The Browse button appears.
2. Click the Browse button. The Region Modeling Tool appears.
3. From the Region menu of the Region Modeling Tool, select Create From Address. The Create from Address screen appears.
 - a. In the Location Name field, enter Oracle.
 - b. In Line 1 of the Address section, enter 500 Oracle Parkway
 - c. In the City field, enter Redwood City.
 - d. In the State field, enter CA.
 - e. In the Postal Code field, enter 94065.
 - f. In the Country field, enter United States.
 - g. Select the Point at Address radio box in the Type of Region section.
 - h. Click Finish to complete the creation of a UDR from an address. The Oracle URD appears in the left frame of the Region Modeling Tool.

Assigning the User-Defined Region to the Master Service

1. In the left frame of the Region Modeling Tool, right click Oracle node.
2. Click Select. The Creating Folder screen reappears, showing the Area ID of the selected node. The Geocoder service is now only available at 500 Oracle Parkway, Redwood City, California.
3. Click Next. The *Selecting an Adapter* screen appears.

4. Select Geocoder from the list of adapters and click Next.
5. Continue clicking the Next button until you reach the confirmation screen.

9.6.2 Task 2: Creating an Alias that Points to Geocoder

Next you can make a master service accessible to a user by placing it in a service tree that a user owns or can access.

1. Right-click the folder, Geocoder in the Wireless Edition repository tree.
2. Click Create New Alias to display the Create New Alias form.
3. Complete this form as follows:
 - a. In the Name field, enter MyGeocoder.
 - b. Click the Browse button next to the Service field. A Browse Services dialog appears. Expand the Master Services folder within the Browse Services dialog and select Geocoder. Click OK.
 - c. Select the Valid check box.
 - d. Select the Visible check box.
4. Click Finish.

9.6.3 Task 3: Making Geocoder Available to a Group

To make Geocoder available to a group:

1. Expand the Groups folder in the Wireless Edition repository tree view.
2. Click MySampleGroup. The parameters for the group appear in the right frame.
3. Click the Service Folders tab in the right frame to display the Service Folders panel. The Service Folders panel includes two fields, Service Folders and Selected Service Folders.
4. Under Service Folders, expand the Services folder.
5. Click Spatial Services.
6. Click the right arrow (>) button to move Spatial Services to the Selected Folders field.
7. Click Apply.
8. Click Yes in the Confirm Apply Changes dialog.

This makes the service available to all users in the group MySampleGroup.

9.6.4 Task 4: Testing the Geocoder

In this walkthrough, you test the Wireless Edition service, Geocoder, using a Web browser and a phone simulator.

Testing the Service on a Browser

Log on to a browser to test the Geocoder service. To access the URL:

1. From a Web browser, enter the following URL:

`http://9iASWEServer.domain/ptg/rm`

Note: The default listener port for the Oracle9i Application Server is 7777. The URL for this port is `http://hostname.domain:7777`. Refer to the Oracle9i Application Server documentation to change the HTTP listener to port 80.

2. In the browser, enter 1234 for the user name and enter 1234 for the password. The services available to user 1234 appear in the left frame.
3. Click Spatial Services, then My Geocoder Services.
4. Click Geocoder. The Geocoder form appears.
5. Complete the Geocoder form as follows:
 - a. Type 500 Oracle Parkway in Address Line 1.
 - b. In the City field, type Redwood City.
 - c. Type CA in the State field
 - d. Type 94065 in the Postal Code field
 - e. Type United States in the Country field.
 - f. Click Submit. A Map pin-pointing the address displays.

Testing Geocoder on a Phone Simulator

You can use any phone simulator (for example, WML, HDML, CHTML) to test Geocoder. If you do not already have one, download and install a simulator. If you are working behind a firewall, you must configure the proxy server settings in the simulator before using it to access external sites.

Follow these steps to test the service from a phone simulator.

1. Start the simulator.
2. Enter the following URL in the Go window:

`http://9iASWEServer.domain/ptg/rm`

This is the URL of the device portal for your Wireless Edition installation.

Note: The default port is 7777. The URL for this port is `http://9iASWEServer.domain:7777/ptg/rm`.

3. Log in using 1234 as the user name and password. The preconfigured Wireless Edition services for user 1234 appear.
4. Select Database Services, then My Database Services.

Note: If the phone simulator returns an HTTP error, you should refresh the cookie cache and source cache. Go to the phone simulator install directory and refresh (or delete) the files **CookieCache** and **SourceCache**.

5. Click Geocoder.
6. Enter the address as you did in [Section 9.6.4](#).

9.7 Creating the Spatial Adapters

You can create your own spatial adapters using the Service Designer.

1. From the Repository Tree, right-click Adapters. The Create New Adapter form appears.
2. Complete the Create New Adapter form as follows:

- a. Enter Geocoder in the Name field.
- b. select the Valid check box.
- c. enter the following in the Class Name field:
`sampleadapter.spatial.geocoder.SampleGeocoder`
- d. Click Finish to complete creation of the Adapter.

Note: You can create another spatial adapter, such as the yellow page adapter, by following the steps described in this section.

9.8 Creating a Master Service Using the Web Integration Adapter

In this walkthrough, you create a Wireless Edition Web service. First, you use the Web Integration Developer to map the elements of a sample Web page, `quote.yahoo.com`, to input and output parameters. You then use the Service Designer to create a Oracle9i Application Server Wireless Edition service that retrieves stock quotes based on the mapping.

For more information on using the Web Integration Developer, see the Oracle9i Application Server Wireless Edition Developers's Guide.

This walkthrough devotes a section to each task associated with creating a master service. These sections include:

- [Section 9.8.1, "Task 1: Creating the WIDL File in the Web Integration Developer"](#)
- [Section 9.8.2, "Task 2: Publishing the WIDL Interface to the Web Integration Server"](#)
- [Section 9.8.3, "Task 3: Using the Service Designer"](#)
- [Section 9.8.4, "Task 4: Creating the Master Service"](#)
- [Section 9.8.5, "Task 5: Creating an Alias that Points to Stock Quotes"](#)
- [Section 9.8.6, "Task 6: Making Stock Quotes Available to a Group"](#)
- [Section 9.8.7, "Task 7: Testing Stock Quotes"](#)

9.8.1 Task 1: Creating the WIDL File in the Web Integration Developer

Follow these steps to create a Web page mapping, or Web Integration Definition Language (WIDL) file, using the Web Integration Developer.

Start the Web Integration Developer

1. Click Start on the Windows NT desktop and point to Programs.
2. From the Programs menu, select Oracle for Windows NT.
3. Select Wireless Edition, then select Web Integration Developer. The Web Integration Developer appears.

Open the Source Page

Open the source page in the Web Integration Developer as follows:

1. Select Open URL from the File menu or toolbar.
2. In the Open URL dialog box, type the following URL:
`http://quote.yahoo.com`

Note: Go to the Configure menu and set proxy to go through the firewall.

3. Click OK.

The Web Integration Developer retrieves the page, parses it, and adds it to the Document Browser (in the left frame). For this example, the Document Browser shows the following items:

- The first item (`http://quote.yahoo.com`) identifies the open document.
- The second item (the Document node) represents the contents of the HTML document: paragraphs, images, links, lists, and tables.
- The third item (the FirstForm node) represents the form in this document. When you open a page that contains forms, the Web Integration Developer creates a form node for each form in the document.

Generate a WIDL File

Next, generate a WIDL file from the source page as follows:

1. Select FirstForm in the left frame.

2. Select WIDL from the Generate menu.
3. Complete the New Service dialog as follows:

Table 9–3 Parameters for the New Service Dialog

In this Field...	Type...
Interface	StockInfo
Service	QuoteYahoo_GetQuote

4. Click OK.
5. In the Generate New WIDL for Service dialog, type ORCL.
6. Click the Submit button.

Edit the Input Binding of the WIDL File

The input variables defined in this WIDL service have the same names as those specified in the HTML form. You can edit the input binding to make these names more meaningful as follows:

1. Expand the Bindings folder in the left frame.
2. Click the QuoteYahoo_GetQuoteInput binding.
3. Click the variable “d” in the variable list in the right frame. In the Name field, type `ReportType` and press Enter.
4. Click the variable “s” in the variable list. In the Name field, type `CoSymbol` and press Enter.

Edit the Output Binding of the WIDL File

The output defined by this service extracts all elements from the document returned by the Web page. To extract just the stock quotes to pass back to the client application, you can edit the output binding:

1. Select the QuoteYahoo_GetQuoteOutput binding in the left frame.
2. In the right frame, scroll through the variable list until you reach the table variables.
3. Click the variable `table8`. Make sure the content of this variable is the stock price of ORCL. If not, click the variable whose content is ORCL’s stock price.
4. On the Sample tab, select the cell that contains the current stock price.

5. Click the right mouse button and select **Create New Variable From Selection** from the pop-up menu.
6. Type `CurrentPrice` in the **New Variable** dialog and click **OK**. The `CurrentPrice` variable now appears in the variable list.
7. Delete all other variables from the variable list.
8. Save the WIDL file.

Test the WIDL File

To test the WIDL file in the Web Integration Developer:

1. Click `QuoteYahoo_GetQuote` in the **Services** folder in the left frame.
2. Select **Test Service** from the **Tools** menu.
3. Type any valid stock symbol in the `CoSymbol` field.

9.8.2 Task 2: Publishing the WIDL Interface to the Web Integration Server

Publishing a WIDL interface makes it accessible to Web Integration services that you create in the Service Designer.

You must have administrator authority on the Web Integration Server to perform this procedure. When you publish an interface, the services in that interface are added to those already on the Web Integration Server. If you create a service with the same name as an existing service, the existing service is overwritten.

Follow these steps to publish your WIDL file (`StockInfo`) to the Web Integration Server.

1. Select `StockInfo` in the left frame.
2. From the **File** menu, select **Publishing**, then **Publish Interface**.
3. In the **Specify Server** field in the **Publish Interface** dialog, type the name of the Web Integration Server to which you want to publish this interface. Specify the server name using the format:
`host_name:port`
4. The Web Integration Server uses packages to organize services. You can click **Update Packages** to view a list of packages on the specified server, then add the service to a specific package. In this case, however, you can add the sample service to the **Default** package. Click **OK**.

5. If the User Name and Password dialog appears, enter a user name and password for the selected server. This user must have administrative privileges. Click OK.

The Web Integration Developer copies the interface to the selected package on the Web Integration Server and notifies you that the interface is successfully published. Assuming the package is enabled, the services in this interface are immediately available for execution.

9.8.3 Task 3: Using the Service Designer

After creating the WIDL service, you create a master service based on the WIDL service. You can create Wireless Edition services using the Service Designer. The Service Designer is a visual tool for creating and managing Wireless Edition objects.

Start the Service Designer

To start the Service Designer, from the Start menu, point to the Oracle for Windows NT menu item. Click Wireless Edition, then Service Designer.

Connect to the Wireless Edition Repository

When the Service Designer starts, the Log In dialog appears. Use the Log In dialog to connect the Service Designer to the Wireless Edition repository that you want to develop. The Log In dialog enables you to select a server name that you have already used, or to enter a new server name.

To connect to the Wireless Edition repository from the Log In dialog:

1. In the location field, enter the name of the Wireless Edition server. Then enter Administrator as the user name and manager as the password.

Note: Both user name and password as case-sensitive.

2. Click OK. The Wireless Edition server objects appear in the tree view on the left.

Note: The Service Designer displays an Error dialog for invalid entries. By clicking the Show Details button, you can view the Java runtime exception details.

9.8.4 Task 4: Creating the Master Service

In this walkthrough, you use the Master Service Creation Wizard to create a Wireless Edition service StockInfo WIDL based on the Web integration adapter. You create a service that displays the current stock price after you enter a valid stock ticker symbol (for example, ORCL).

Before you begin the walkthrough, make sure you can access a Wireless Edition server from your development environment running the Service Designer.

Using the Master Service Creation Wizard

Start the Service Designer and follow these steps:

1. Click the Tools menu in the Service Designer and select Master Service Creation Wizard. The Welcome screen of the Master Service Creation Wizard appears which lists the necessary steps for the creation of a master service. Click Next to go to the Selecting or Creating a New Service Folder screen.
2. Select the Create a New Service Folder radio button.
3. Click Next. The Creating Folder screen appears. Enter the following information in the In the New Folder Info section:
 - a. For the Name field, enter My Folder.
 - b. Select the Valid check box.
 - c. Select the Visible check box.
 - d. Enter a description of the service that is meaningful to you, such as Stock Prices.
4. Click Next. The Selecting or Creating a Service Folder screen reappears. Click My Folder and click Next. The Setting the Master Service Properties screen appears. Enter the following information in the Master Service Info section:
 - a. For the Name field, enter Stock Quotes.
 - b. Select the Valid check box.
 - c. Select the Visible check box.
5. Click Next. The Selecting an Adapter screen appears. Select WebIntegrationAdapter from the list of adapters and
6. Click Next. The screen for Setting the Init Parameters appears. Enter the following information into the fields.

- a. For the Web Integration Server, type the machine name of the Web Integration Server in the format: *host_name.domain:port*
 - b. For the Interface field, type `StockInfo`. This is the name of the WIDL interface you created in [Section 9.8.1](#)
 - c. Do not supply a value for the `WIDL_FILE` parameter.
7. Click Apply. The Setting the Input Parameters screen appears. The Input parameters screen is populated with the input parameters for the `WebIntegrationAdapter`. Select the User Customizable check box for the `CoSymbol` parameter.
 8. Click Next to go to the Setting the Output Parameters screen. The Output parameters screen is also populated with the output parameters for the `WebIntegrationAdapter`. You do not need to modify any of the output parameters.
 9. Click Finish. You have created a master service.

This creates the master service `Stock Quotes` in the Oracle9i Application Server Wireless Edition repository. You can now create aliases to make the master service available to users, assign the service to the group, and test it.

9.8.5 Task 5: Creating an Alias that Points to Stock Quotes

Using the Service Designer, you can make a master service accessible to a user by placing it in a service tree that a user owns or can access. To create an alias for the service `Stock Quotes`:

1. Right-click My Folder.
2. Click Create New Alias to display the Create New Alias form.
3. Complete this form as follows:
 - a. In the Name field, enter My Services.
 - b. Click the Browse button next to the Service field. A Browse Services dialog appears. Expand the Master Services folder within the Browse Services dialog and select `Stock Quotes`. Click OK.
 - c. Select the Valid check box.
 - d. Select the Visible check box.
4. Click Finish.

5. Click the Input Parameters tab in the right frame and select the User Customizable check box for CoSymbol.

This creates an alias object in the Wireless Edition repository. Aliases, like master services, have input parameters. You can use the input parameters of an alias to override any input values specified by the master service. For this walkthrough, you use the default settings provided by the master service.

9.8.6 Task 6: Making Stock Quotes Available to a Group

You use the Service Designer to make the service you created available to a group of users:

1. Expand the Groups folder in the Wireless Edition repository tree view.
2. Click MySampleGroup. The parameters for the group appear in the right frame.
3. Click the Service Folders tab in the right frame to display the Service Folders panel. The Service Folders panel includes two fields, Service Folders and Selected Service Folders.
4. Under Service Folders, expand the Services folder.
5. Click My Folder.
6. Click the right arrow (>) button to move Stock Quotes to the Selected Folders field.
7. Click Apply.
8. Click Yes in the Confirm Apply Changes dialog.

This makes the service available to all users in the group MySampleGroup. In [Section 9.8.7](#), you test this service as the user 1234.

9.8.7 Task 7: Testing Stock Quotes

In this walkthrough, you test the Wireless Edition service, Stock Quotes, using a browser and a phone simulator.

Testing the Service on a Browser

Log on to a browser to test the service. To access the URL:

1. From a Web browser, enter the following URL:
`http://9iASWEServer.domain/ptg/rm`

Note: The default listener port for the Oracle9i Application Server is 7777. The URL for this port is `http://9iASWEServer.domain:7777`. Refer to the Oracle9i Application Server documentation to change the HTTP listener to port 80.

2. Enter 1234 for the user name and enter 1234 for the password. The services available to user 1234 appear in the left frame.
3. Click My Folder, then My Services.
4. Click QuoteYahoo GetQuote
5. Enter a valid stock symbol and click submit.

The Wireless Edition retrieves and displays the current stock price.

Testing Stock Quotes on a Phone Simulator

You can use any phone simulator (for example, WML, HDML, CHTML) to test Stock Quotes. If you do not already have one, download and install a simulator. If you are working behind a firewall, you must configure the proxy server settings in the simulator before using it to access external sites.

Follow these steps to test the service from a phone simulator.

1. Start the simulator.
2. Enter the following URL in the Go window:

`http://9iASWEServer.domain/ptg/rm`

This is the URL of the device portal for your Wireless Edition installation.

Note: The default port is 7777. The URL for this port is `http://9iASWEServer.domain:7777/ptg/rm`.

3. Log in using 1234 as the user name and password. The preconfigured Wireless Edition services for user 1234 appear.

4. Select My Folder, then My Services.

Note: If the phone simulator returns an HTTP error, you should refresh the cookie cache and source cache. Go to the phone simulator install directory and refresh (or delete) the files **CookieCache** and **SourceCache**.

5. Enter a valid stock ticker symbol (for example, ORCL) and click OK.

The Wireless Edition retrieves and displays the current price of the stock.

9.9 Creating Chained Services Using the Web Integration Developer

Chained services link services to each other in a sequential way so that each chained service is dependent for input on the service that immediately precedes it. Chained services are needed for Web sites which require the user to interact with a Web page to access the rest of the content on the Web site. For example, the first Web page prompts the user for a user name and password before the user can access the form on the second page, which then produces the information on the third page. Chaining one service (Service A) to another service (Service B) works in such a way that when the Output Binding of Service A completes successfully, it chains to the Input Binding of Service B.

The chained service you build in this walkthrough illustrates a typical use of chained services. Based on the Yahoo finance site, the chained service you build in this section allows the user to enter a stock ticker symbol, receive the current stock quote, select details, and receive detailed information about the stock, such as a chart. The first service requests that the user enter a stock ticker symbol. It returns the current quote for that stock. When the user clicks on Details and selects Charts, the second service returns the chart for that stock.

You should become familiar with a site you want to base a service on by first opening it in a browser. Opening the site and viewing its HTML source code lets you become familiar with the structure of a site. In a browser, open the Yahoo site for getting stock quotes:

<http://quote.yahoo.com>

The query form on this page takes a stock's ticker symbol. It returns that stock's current quote.

This walkthrough assumes that you are familiar with the basic steps required to create a Web service in the Oracle Web Integration Developer. If you are not,

complete the Web Integration walkthrough before attempting to follow this walkthrough.

This walkthrough does not include error checking. Usually, you would add error catching logic to the service.

9.9.1 Task 1: Creating the First Service in the Chain

The overall interface name for the chained services in this walkthrough is YahooFinance. The first service in the chain sequence is called Quote. This service prompts the user for a stock ticker symbol (such as, ORCL) and returns the stock quote. The following steps provide a basic method for creating the first service in the chain.

Generate the WIDL File and Edit the Input Binding

1. In the Oracle Web Integration Developer, open the following URL:

<http://quote.yahoo.com>

Two nodes appear in the document browser, Document and FirstForm.

2. Highlight the FirstForm node and click Generate, then WIDL. Complete the New Service dialog as follows and click OK:

Table 9–4 Parameters of the New Service Dialog

In this Field...	Type...
Interface	YahooFinance
Service	Quote

3. In the Generate a New WIDL for Service "Quote" dialog box, enter the stock symbol ORCL. Click Submit.
4. Expand the bindings node and highlight the QuoteInput binding by selecting it. The variables in the Quote service's input binding appear in the edit pane of the Variables tab.
5. Rename the variable s to s1, and d to d1.

Edit the Output Binding of the WIDL File

1. Select the QuoteOutput binding. In the variables list, highlight the row for table4 by clicking on it. The Sample tab shows the values in table4. You can create new variables based on these values.
2. In the Sample tab, highlight the ORCL cell. Right click. Select **Create new variable from Selection**. Name the new variable **symbol**.
3. Similarly, highlight the cell that shows the ORCL stock price (in the table4 sample result). Right click. Select **Create new variable from Selection**. Name the new variable **price**.
4. Follow the same procedure as in the above two steps to create the new variable **change** from the cell that shows the value for change.
5. Delete all output variables from QuoteOutput, except for the three new output variables (symbol, price, change) that you just created in steps 7, 8, and 9 above.
6. Save the WIDL file.
7. Test the first service, Quote, with other stock symbols.

9.9.2 Task 2: Creating the Second Service in the Chain

This walkthrough takes you through the steps to build the second service, named **Detail**, in the chained service YahooFinance. While you can create chained services using separate interfaces, in this walkthrough you add the second service to the YahooFinance interface.

The Detail service takes the stock quote as input and returns stock charts.

Create the second service as follows:

Open the Source Page for the Second Service

1. Open the following page in a browser:
`http://finance.yahoo.com`
2. Enter ORCL as the quote symbol and select Chart as the type of search (instead of Basic).
3. Click Search. As you see, a chart search returns more information than a basic search. The second service in the chain returns this detailed information.
4. Select the URL in your browser's Location field,
`http://finance.yahoo.com/q?s=orcl&d=b`, and open this URL in the Web Integration Developer.

Generate the WIDL for the Second Service

1. Select FirstForm in the left frame.
2. Select WIDL from the Generate menu.
3. Complete the New Service dialog as follows:

Table 9–5 Parameters for the New Service Dialog

In this Field...	Type...
Interface	YahooFinance
Service	Detail

4. Click OK.
5. In the Generate a New WIDL for Service "Detail" dialog box, enter the stock symbol ORCL. Click the Submit button.

Edit the Input and Output Binding of the WIDL File

1. Expand the Bindings folder in the left frame.
2. Select DetailInput. Rename the variable s to s2, and d to d2.
3. Select DetailOutput. In the variables list, highlight table4.
4. Find the cell with the Open price. Highlight the cell, right click and create a new variable from selection. Name the new variable **opened**.
5. From the day's range cell, create a new variable named **range**.
6. Similarly, from the cell showing volume, create a new variable named **volume**.
7. From the cell showing the 52-week range, create a new variable named **annualrange**.
8. Create a new variable named **pe** from the cell that shows the P/E value.
9. Create a new variable named **mktcap** from the cell that says Mkt cap.
10. Delete all variables other than the ones that you just created in steps 9 to 14 above.

Edit the URL of the Second Service

1. Highlight **Detail** under the Services node.
2. Remove q?s=orcl&d=b from the Source URL.

3. Save the WIDL file.

Publish the Service

Publish the new service to the WIDL server, as described in [Section 9.8.2](#)

9.9.3 Adding the Service to the Wireless Edition Repository

The new Master Service is YahooFinance. It points to the interface YahooFinance; it includes the two chained services **Quote** and **Detail**. To add this service to the Wireless Edition repository, start the Service Designer and follow these steps:

Using the Master Service Creation Wizard

Start the Service Designer and follow these steps:

1. Click the Tools menu in the Service Designer and select Master Service Creation Wizard. The Welcome screen of the Master Service Creation Wizard appears. Click Next to go to the Selecting or Creating a New Service Folder screen.
2. From the Selecting or Creating a New Service Folder screen, select My Folder, which you created in [Section 9.8.4](#).
3. Click Next. The Setting the Master Service Properties screen appears. Enter the following information in the Master Service Info section:
 - a. For the Name field, enter YahooFinance.
 - b. Select the Valid check box.
 - c. Select the Visible check box.
4. Click Next. The Selecting an Adapter screen appears. Select WebIntegration Adapter and then click Next. The Screen for setting the Init Parameters appears. Enter the following information:
 - a. For the Web Integration Server, type the machine name of the Web Integration Server in the format: *host_name.domain:port*
 - b. For the Interface field, type YahooFinance.
5. Click Apply. The Setting Input Parameters screen appears, populated with the input parameters for the WebIntegrationAdapter. Complete the screen as follows:
 - a. Select the User Customizable check box for s1.

- b. Click the Value field of the PAsection variable. Select Quote from the drop-down menu.
 - c. For the variable OutputType, set a value of RawResult.
 - d. Click the Add button. Click in the Name field of the new input variable and type PAdebug. Set the value to 1.
 - e. Deselect the User Customizable check box for PAsection, if set.
6. Click Next. The Setting the Output parameters screen appears, populated by the output parameters of the WebIntegrationAdapter.
 7. Click the Add button. In the Name field, delete the default name, and enter Quote.
 8. Click Next. The Creating the Result Transformer screen appears. Paste the following style sheet in the XSL Editor window and click Next. Be sure to replace the existing template text in the window.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <xsl:apply-templates
select="ServiceRequest/Result/AdapterResult"></xsl:apply-templates>
  </xsl:template>
  <xsl:template match="AdapterResult">
    <SimpleResult>
      <SimpleMenu>
        <SimpleMenuItem>
          <xsl:attribute
name="target">__REQUEST_NAME__?PAservicepath=__SERVICE_URL_ENC__
&#38;PAsection=Detail&#38;s2=<xsl:value-of
select="symbol"></xsl:value-of></xsl:attribute>
          <xsl:value-of select="symbol"></xsl:value-of>
          <!-- this xsl:text is just for beautifying the output -->
          <xsl:text> - </xsl:text>
          <xsl:value-of select="price"></xsl:value-of>
          <!-- this xsl:text is just for beautifying the output -->
          <xsl:text> - </xsl:text>
          <xsl:value-of select="change"></xsl:value-of>
        </SimpleMenuItem>
      </SimpleMenu>
    </SimpleResult>
  </xsl:template>
</xsl:stylesheet>
```

9. Click the Add button. A name field for a new stylesheet appears with a default name.
10. Replace the default name with Detail. Click the Edit button.
11. Replace the existing template with the following and click OK:

```
<xsl:stylesheet version = "1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <xsl:apply-templates
select="ServiceRequest/Result/AdapterResult"></xsl:apply-templates>
  </xsl:template>
  <xsl:template match="AdapterResult">
    <SimpleResult>
      <SimpleContainer>
        <SimpleText>
          <SimpleTextItem>
            <xsl:value-of select="opened"></xsl:value-of>
          </SimpleTextItem>
          <SimpleTextItem>
            <xsl:value-of select="range"></xsl:value-of>
          </SimpleTextItem>
          <SimpleTextItem>
            <xsl:value-of select="volume"></xsl:value-of>
          </SimpleTextItem>
          <SimpleTextItem>
            <xsl:value-of select="annualrange"></xsl:value-of>
          </SimpleTextItem>
          <SimpleTextItem>
            <xsl:value-of select="pe"></xsl:value-of>
          </SimpleTextItem>
          <SimpleTextItem>
            <xsl:value-of select="mktcap"></xsl:value-of>
          </SimpleTextItem>
        </SimpleText>
      </SimpleContainer>
    </SimpleResult>
  </xsl:template>
</xsl:stylesheet>
```

9.9.4 Creating a User and an Alias and Test the Service

1. Create a new user and a new alias as described [Section 9.8.5](#).
2. Test the service from the Personalization portal as described in [Section 9.8.7](#)

9.10 Creating External Links Using the Service Designer

An External Link is a Wireless Edition service that points to an external resource. The external resource is typically a Web page that serves content in a format supported by the target device. When an end user sets an external link, it appears as a menu selection in the device portal.

The Wireless Edition does not process the content of the external link target. As a result, external link services are not available to all targeted devices, as are other Wireless Edition services.

In this walkthrough, you will create an external link using the Service Designer.

To create an external link in the Service Designer:

1. Highlight and right-click Database Services in the Wireless Edition repository tree.
2. Click Create New Bookmark. The Create New Bookmark form appears.

Figure 9–2 The Create New Bookmark Form

The screenshot shows a dialog box titled "Create New Bookmark". It contains the following fields and controls:

- Name:** A text input field.
- Valid:** A checkbox, currently unchecked.
- Visible:** A checkbox, currently unchecked.
- Owner:** A text input field containing "Administrator" and a "Browse" button to its right.
- HREF:** A text input field.
- Seq No:** A small text input field.
- Description:** A large text input field.
- Buttons:** "Cancel" and "Finish" buttons at the bottom.

3. Complete the Create New Bookmark form as follows:

- a. In the Name field, enter My Oracle.

Note: The name of the bookmark must be unique within its parent folder.

- b. Select the Valid check box.
 - c. Select the Visible check box.
 - d. Click Browse to select user 1234 as the owner of the external link.
 - e. Enter `http://www.oracle.com` as the HREF.
 - f. Enter a description that is meaningful to you and the external link's user.
4. Click Finish. The external link, My Oracle, appears in the Wireless Edition repository tree under the Database Services folder.

The Mail Adapter

This appendix describes the Mail Adapter. Each section of this appendix presents a different topic. These sections include:

- [Section A.1, "Overview"](#)
- [Section A.2, "The End User Interface"](#)
- [Section A.3, "Configuration Parameters for the Mail Adapter"](#)

A.1 Overview

The Mail adapter is based on the JavaSoft's Java Mail API and provides mail services to both POP3 and IMAP4 mail servers. It provides basic authentication and mail session management. The Mail adapter enables users to browse, create, delete, reply, and forward mail and folders from their wireless devices. The Mail adapter also provides user interfaces for these tasks.

A.1.1 Message Management Functions

The email adapter enables the following message management functions:

- Read each individual email message. The email adapter delivers the plain text portion to the mobile device. The mail adapter does not include attachments to messages.
- Manage messages with large content. The adapter can deliver partial content, scroll down to the next portion of the message, or scroll up to the previous portion.
- Delete selected email.
- Reply to selected email.

- Move selected email to a different folder.
- Compose a new email message.

A.1.2 Folders Management Functions

The Mail adapter can perform the following folder management tasks:

- Create a Folder
- Delete a folder.
- Manage folders containing large numbers of messages.
- View all the messages in a folder.

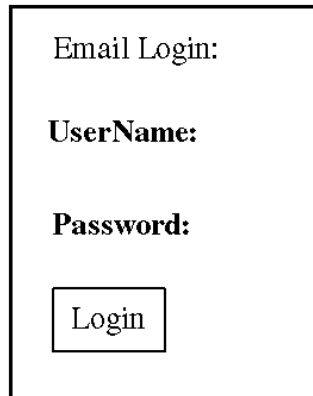
Note: Developers can modify the mail adapter by adding functions.

A.2 The End User Interface

The Mail adapter includes an interface that provides users with a sequence of pages for executing messaging functions. The pages are as follows:

The Login Page

The Login page prompts the user name and password. The Read Inbox Folder page appears after a user has successfully logged in.

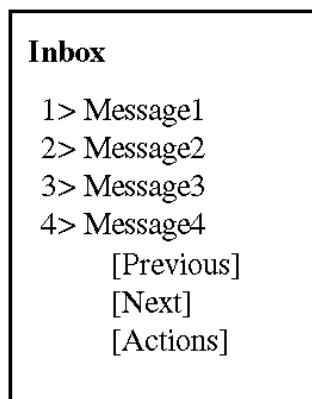
Figure A-1 The Login Page

The screenshot shows a login form with the following elements:

- Email Login:
- UserName:**
- Password:**
- Login

The Inbox Page

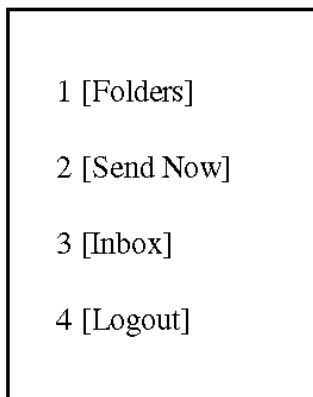
The Inbox page displays the users current email inbox and enables users to access their messages or navigate to other pages using the Previous and Next options.

Figure A-2 The Inbox Page

The screenshot shows an inbox page with the following elements:

- Inbox**
- 1> Message1
- 2> Message2
- 3> Message3
- 4> Message4
- [Previous]
- [Next]
- [Actions]

When users select the Actions option, they go to the Actions Screen.

Figure A–3 The Actions Page**The Actions Page**

The Actions page presents users with the following options:

Table A–1 Options in the Actions Page

Option	Description
Folders	Selecting this option moves users to the Folder Listing page.
Send New	Selecting this option moves users to the Compose Message page.
Inbox	Selecting this option moves users to the Inbox page.
Logout	Selecting this options enables users to exit the service.

The Compose Message Page

Users create messages by selecting the Send New option from the Action page. The Compose Message page presents users with a form that requests the recipient of the message, the subject of the message and the message itself. After users complete this form, they select the Submit option to transmit the message.

Figure A-4 The Compose Message Page

Please fill in the form below.

Receiver

Subject

Message

[Submit]

The Folder Listing Page

When users select the Folders option from the Action screen, they move to the Folder Listing screen. Clicking one of the folders listed in this screen displays a corresponding Folder screen, which lists messages contained within that folder. Selecting the Back option takes users back to the Action screen. Selecting the Create Fold option takes users to the Create Folder Screen.

Figure A-5 The Folder Listing Page

Folders

1 Folder1

2 Folder2

3 Folder3

4 Folder4

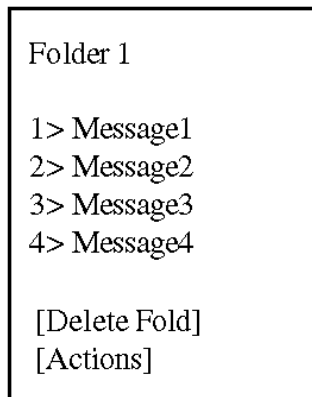
[Back]

[Create Fold]

The Folder Page

From the Folder page, users can then access their messages by selecting the listed message items. Selecting any of the listed message items moves users to the Read Message page. The Folder Listing page also enables users to delete the folder by selecting the Delete Fold option, or to navigate to the Action screen using the Actions option.

Figure A-6 *The Folder Page*

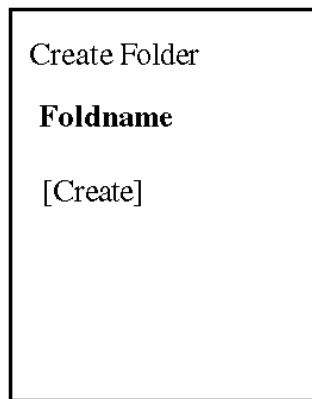


Note: Developers specify the number of messages displayed in this screen in **mailadapter.properties**. For more information, see the *Oracle9i Application Server Wireless Edition Configuration Guide*.

The Create Folder Page

When users select the Create Fold option in the Folder Listing page, they move to the Create Folder page. This page enables user to create a new folder by entering a folder name and then by selecting the Create option. The folder created using this page is at the same level as the inbox folder.

Figure A-7 The Create Folder Page



Create Folder

Foldname

[Create]

The Read Message Page

Selecting from among the listed messages in the Folder page moves users to the Read Message page. This page displays the sender, date, and subject of the message, as well as the message itself. The Read message page enables users to navigate back to the Action page by selecting the Back option, reply to the message by selecting the Reply option, or forward the message by selecting the Forward option. In addition, users can move the message to another folder using the Move option, and can delete the message by selecting the Delete option. When users chose to delete a message, a confirmation page appears asking them if they want to proceed with the deletion.

Figure A-8 The Read Message Page

From:
scott.tiger @oracle.com
Date: Jan 10, 2000
Subject: Movie

Do you want to see a
movie tonight?

1 [Back]
2 [Reply]
3 [Forward]
4 [Move]

The Reply Page

Users invoke the Reply page when they select the Reply option in the Read Message page. To reply to a message, users enter their message and then select the Submit option to transmit the message.

Figure A-9 The Reply Message Page

Please fill in the form
below.

Receiver
Scott.Tiger@oracle.com

Subject
Re: Movie

Message
Sure!

[Submit]

The Forward Message Page

Users invoke the Forward Message page when they select the Forward option in the Read Message page. To forward a message, users enter the recipient of the forwarded message, and text of the message. Users then select the Submit option to transmit the message.

Figure A-10 The Forward Message Page

Please fill in the form below.

Receiver
user.one@oracle.com

Subject
forward: Movie

Message
Sure!

[Submit]

The Move Message Page

Users invoke the Move Message page when they select the Move option in the Read Message page. To move the message displayed in the Read Message screen to another folder, users first select the Move option. The Move Message page appears, displaying a list of folders. Users then select a folder.

Figure A–11 The Move Message Page

Pick one Folder

1 Folder1
2 Folder2
3 Folder3
4 Folder4

A.3 Configuration Parameters for the Mail Adapter

Because mobile devices have limited display and memory, the following configuration parameters are used to control how many messages and how much content each message can display at a one time on a mobile device. The configuration parameters are defined in **mailadapter.properties** in `oracle\panama\adapter\mailadapter\`.

Maximum Number of Messages Listed in a Screen

`JavaMail.User_DISPLAY_NUM=6`

Maximum Number of Characters Sent to a Device in Each Trip to the Device

`JavaMail.EMAIL_CONTENT_NUM=100`

Glossary

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 Wireless Edition 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.

bookmark

A link from a service to an external, device-compatible data source that does not require Wireless Edition processing.

core

The Wireless Edition component that manages the Wireless Edition repository and service requests.

daemon

A background process that performs a specified operation in response to certain events or at specified times.

device portal

The interface where mobile device users access their Wireless Edition services.

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 Wireless Edition service from a client device.

filtering

The process of transforming content by replacing existing markup tags with tags that represent another format.

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.

logical 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 logical device.

master service

The core implementation of a service. The master service object invokes a specific adapter, and identifies the transformer used to convert content for the target device.

MIME

Multipurpose Internet Mail Extensions. A mail type that defines the message structure for different 8-bit character sets and multi-part messages.

Personalization Portal

A Web-based interface that end users access to select services and configure their device portal. Users access the Personalization Portal from their desktop computers.

private service tree

A service tree that is owned by a specific end user.

provisioning adapter

The adapter used to create, modify, and delete user objects in the Wireless Edition repository.

public service tree

A service tree that is owned by the Wireless Edition system, and can be accessed by any end-user.

repository

An Oracle8i database which stores all of the Wireless Edition objects, such as users, groups, adapters, and services.

request

A query to initiate a desired Wireless Edition service. Requests are submitted on behalf of end-users to the Wireless Edition server.

request manager

The Wireless Edition component that processes requests for services. The request manager authenticates the user, submits the request to the Wireless Edition 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 into Simple Result format.

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 Wireless Edition repository, which includes pre-built objects such as transformers, adapters, and logical devices.

service

A core object used in a Wireless Edition server to represent a unit of information requested by, and delivered to, a Wireless Edition client. An end user typically sees a service as a menu item on a device or as a link on a Web page.

service alias

A pointer to a master service. When a service alias is placed in a service tree, the corresponding service becomes available to the owner or owners of the service tree.

service designer

The visual interface for creating and managing Portal-to-Go users, user groups, adapters, transformers, and services.

service tree

A tree data structure containing one or more services. Service trees make services available to end users.

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 Wireless Edition adapter. For example, the source format of Web page content is HTML.

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.

SQL adapter

An adapter that retrieves and adapts content from any JDBC-enabled data source.

stylesheet

An XSLT (eXtensible Stylesheet Language Transformations) instance that implements content presentation for XML documents. The Wireless Edition transformers can be either XSLT stylesheets or Java programs.

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 Wireless Edition repository. Thin HTML does not include support for frames, JavaScript, or other advanced features.

transformer

A Wireless Edition object that converts content returned by the Wireless Edition adapters. Result transformers convert Adapter Result documents into Simple Result documents. Device transformers convert Simple Result documents into the target format.

TTML

Tagged Text Mark-up Language. A lightweight version of HTML suitable for most PDAs.

user agent

An object that associates an end user with a device type.

user group

A Wireless Edition 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 Wireless Edition 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 Wireless Edition service.

Wireless Edition XML

A set of DTDs and XML document conventions used by the Wireless Edition to define content and internal objects.

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, such as product, sales representative, or amount due, can be created and used in specific applications, allowing Web pages to function like database records.

XSLT

Extensible Stylesheet Language Transformations. A stylesheet format for XML documents.

Index

A

Adapter Result format, 3-29
adapters, 1-6, 1-7, 2-2
 description of, 1-6
alerts
 creating from a device, 7-7
 creating with the Personalization Portal, 6-11
 description of, 1-5
aliases, 2-3, 3-3, 3-38
 creating, 3-38, 9-6, 9-15, 9-29, 9-33, 9-42
authenticate method, 5-24
authenticating users, 5-24
autolocate
 activating from a device, 7-8

B

beginTag parameter, 3-31
bookmarks, 2-3
 creating from a device, 7-4
 creating with the Personalization Portal, 6-10

C

caption, parameter property, 3-15, 3-16, 3-27
chained services, 3-30
chained services, creating, 9-45
checkUser method, 5-24
client device type
 recognizing, 8-6
connect string
 storing, 9-2
custom adapter, 1-8

D

device recognition, 8-6
device transformer
 specifying in a master service, 3-34
devices
 activating autolocate, 7-8
 creating alerts on, 7-7
 creating bookmarks on, 7-4
 creating folders on, 7-3
 creating locationmarks on, 7-8
 customizing portals on, 7-2
 editing user profiles on, 7-7
 logging in, 7-3
devices, supporting new, 8-7
Directory adapter, 1-9

E

endTag parameter, 3-32
error page parameter, 8-5
external links
 creating, 9-52

F

folders, 2-3, 3-3
 creating, 3-35
 creating from a device, 7-3
 deleting, 3-37
 editing with the Personalization Portal, 6-8

G

general parameters, master services, 3-21
Geocoding adapter, 1-8

H

HTTP error, when using a phone simulator, 9-18, 9-35, 9-45
HTTP user agent parameter, 8-6

I

implementation walkthroughs, 9-2
initialization parameters, adapter, 3-21
input parameters, adapter, 3-26

L

linked services, 9-45
location-based service (LBS) adapter, 1-9
location-based services, 3-2
 creating, 3-19
 description of, 1-6
locationmarks
 creating from a device, 7-8
 creating with the Personalization Portal, 6-16
 editing with the Personalization Portal, 6-19
logical devices, 2-3, 8-1
 deleting, 8-6
 login page, 8-4
 modifying, 8-5
 parameters, 8-2
login page for a device, 8-4

M

Mail adapter, 1-9
 init parameters of, 3-25
 input parameters of, 3-30
Master Service Creation Wizard, 3-8
master services, 2-2, 3-2
 chained, 3-30
 creating, 9-41
 deleting, 3-35
 description of, 1-9

general panel parameters, 3-21
init parameters, 3-21
input parameters, 3-26
modifying, 3-21

MIME type parameter, 8-2

O

Oracle Performance Manager, 2-9
Oracle XML Version 2 Parser, 2-8
output parameters, 3-34
OutputType parameter, 3-29

P

PAsection parameter, 3-16, 3-17, 3-28
Personalization Portal, 2-8
 Administration tab, 5-7
 finding users with, 5-10
 logging in as a user, 6-2
 logging in as administrator, 5-7
 used on a device, 7-2
Personalization portal
 creating new users with, 5-11
Personalization Portal
 activating service for users, 6-5
phone simulator, using a, 9-8, 9-17, 9-30, 9-35, 9-44
PL/SQL procedures in services, 3-23
property files
 rereading, 8-7
provisioning adapter, 1-9
provisioning systems, integrating, 5-23
ProvisioningHook interface, 5-23

Q

query form, user, 5-2

R

recognition, device, 8-6
Region Modeling Tool, 2-6, 4-1
 accessing, 4-6
 browse mode, 4-17
 creating location-based services with, 4-18

- standalone mode, 4-6
- repository connect string, 9-2
- result transformer
 - specifying in a master service, 3-34
- Routing adapter, 1-8

S

- Service Designer, 2-1
 - creating objects with, 3-6
 - deleting objects with, 3-7
 - functions of, 3-3
 - modifying objects with, 3-6
 - starting, 9-2
- service trees, 3-37
- Servlet adapter, 1-7, 3-24
 - input parameters of, 3-33
- SQL adapter, 1-7
 - initialization parameters, 3-22
 - input parameters, 3-31
- starting the Service Designer, 9-2
- STATEMENT parameter
 - sample, 9-6, 9-14
- stored procedures, calling in services, 3-23
- stripLevel parameter, 3-32
- stripper adapter, 1-7, 3-22
 - input parameters, 3-31
- system-defined region
 - creating, 4-12
 - deleting, 4-17
- system-defined region (SDR), 4-2

T

- testing services, 9-8, 9-16, 9-30, 9-34, 9-43
- Transformer Testing Tool, 2-4
- transformers, 2-3

U

- URL adapter, 1-7, 3-23
 - input parameters of, 3-32
- URL adapter parameter, 3-31
- URL caching, specifying for devices, 8-3
- user agents, 8-6

- User Customizable, parameter property, 3-15, 3-17, 3-27
- user groups, 2-3, 5-2
 - adding users using the Service Designer, 5-4
 - assigning services using the Personalization Portal, 5-22
 - creating with the Service Designer, 5-2
 - deleting using the Personalization Portal, 5-20
 - deleting using the Service Designer, 5-4
 - parameters, 5-2
 - service access, 5-4
- useragent.properties, 8-6
- UserAuthenticationHook interface, 5-24
- user-defined region
 - creating, 4-7
 - deleting, 4-10
- user-defined region (UDR), 4-6
- users
 - assigning to groups, 5-18
 - authenticating, 5-24
 - creating, 5-11, 9-4
 - editing user profiles from a device, 7-7
 - editing user profiles with the Personalization Portal, 6-20
 - integrating provisioning systems, 5-23
 - removing users using the Service Designer, 5-5

W

- walkthroughs, 9-1
- Web Integration adapter, 1-8, 3-22
 - initialization parameters, 3-22
 - input parameters, 3-28
- Web Integration Developer, 2-7
- Web Integration Server, 9-39
- Web Interface Definition Language files., 9-37
- WIDL Interface parameter, 3-22
- WIDL services
 - creating, 9-37
 - publishing, 9-39
- Wireless Edition repository, 2-2
 - RMI connections and, 3-5
- Wireless Edition XML, 1-4

X

XML Editor, 2-7

Y

Yellow Page adapter, 1-8