Oracle® Retail Mobile Point-of-Service

Operations Guide

Release 12.0

September 2007

ORACLE®

Oracle Retail Mobile Point-of-Service Operations Guide, Release 12.0

Primary Author: Graham Fredrickson

Value-Added Reseller (VAR) Language

(i) the software component known as **ACUMATE** developed and licensed by Lucent Technologies Inc. of Murray Hill, New Jersey, to Oracle and imbedded in the Oracle Retail Predictive Application Server - Enterprise Engine, Oracle Retail Category Management, Oracle Retail Item Planning, Oracle Retail Merchandise Financial Planning, Oracle Retail Advanced Inventory Planning and Oracle Retail Demand Forecasting applications.

(ii) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(iii) the **SeeBeyond** component developed and licensed by Sun MicroSystems, Inc. (Sun) of Santa Clara, California, to Oracle and imbedded in the Oracle Retail Integration Bus application.

(iv) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Store Inventory Management.

(v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by Business Objects Software Limited ("Business Objects") and imbedded in Oracle Retail Store Inventory Management.

(vi) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(vii) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

(viii) the software component known as **Style Report™** developed and licensed by InetSoft Technology Corp. of Piscataway, New Jersey, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(ix) the software component known as **i-net Crystal-Clear™** developed and licensed by I-NET Software Inc. of Berlin, Germany, to Oracle and imbedded in the Oracle Retail Central Office and Oracle Retail Back Office applications.

(x) the software component known as **WebLogic™** developed and licensed by BEA Systems, Inc. of San Jose, California, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(xi) the software component known as **DataBeacon™** developed and licensed by Cognos Incorporated of Ottawa, Ontario, Canada, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

Contents

List of Figures

List of Tables

Preface

Oracle Retail Operations Guides contain the requirements and procedures that are necessary for the retailer to configure Mobile Point-of-Service, and extend code for a Mobile Point-of-Service implementation.

Audience

The audiences for this document are administrators and developers who install and configure the Oracle Retail Mobile Point-of-Service application.

Related Documents

For more information, see the following documents in the Oracle Retail Mobile Point-of-Service Release 12.0 documentation set:

- *Oracle Retail Mobile Point-of-Service Release Notes*
- *Oracle Retail Mobile Point-of-Service Installation Guide*
- *Oracle Retail Mobile Point-of-Service User Guide*

Customer Support

- https://metalink.oracle.com

When contacting Customer Support, please provide:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to recreate
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

For a base release (".0" release, such as 12.0), Oracle Retail strongly recommends that you read all patch documentation before you begin installation procedures. Patch documentation can contain critical information related to the base release, based on new information and code changes that have been made since the base release.

Oracle Retail Documentation on the Oracle Technology Network

In addition to being packaged with each product release (on the base or patch level), all Oracle Retail documentation is available on the following Web site:

http://www.oracle.com/technology/documentation/oracle_retail.html

Documentation should be available on this Web site within a month after a product release. Note that documentation is always available with the packaged code on the release date.

Conventions

The following text conventions are used in this document:

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

1

Technical Architecture

Mobile Point-of-Service provides wireless access to a subset of the Oracle Retail Point-of-Service functions. Mobile Point-of-Service enables a cashier to use a handheld device to check out customers. This chapter introduces you to Mobile Point-of-Service. After reading this chapter, you should be able to:

- Log on and off
- Navigate the application

Mobile Point-of-Service uses the same tour framework as the Point-of-Service client to control the process flow of the user interface.

Mobile Point-of-Service also uses the same manager/technician framework as Point-of-Service.

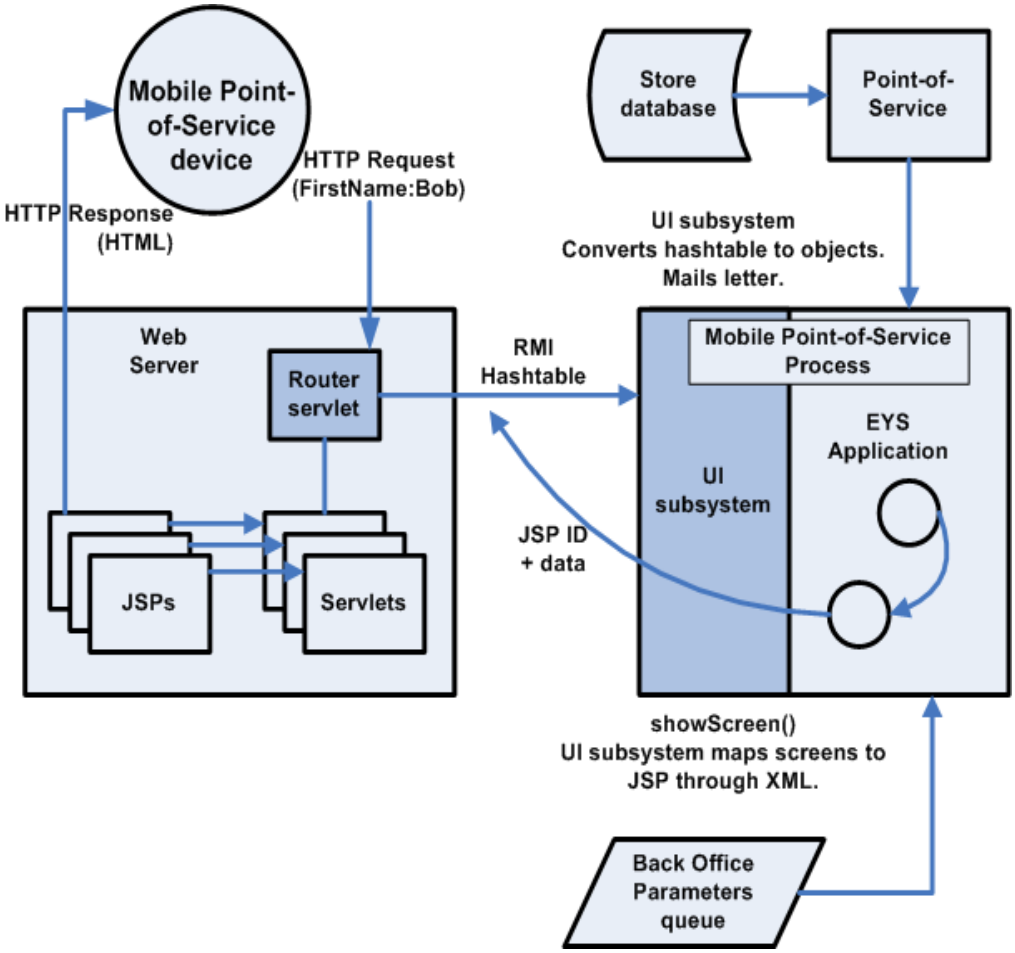
For more information, refer to the Oracle Retail Mobile Point-of-Service Operations Guide.

Architecture

Mobile Point-of-Service is a server-centric Java application. The following is an overview of the communication process:

1. The browser in the handheld unit communicates with a Java servlet that runs on a Web server.
2. The Java servlet communicates with Mobile Point-of-Service.
3. Mobile Point-of-Service changes its state based on the type of user input.
4. Mobile Point-of-Service selects the next screen to display, and invokes another Java servlet to dynamically create an HTML page.
5. The Java servlet delivers the HTML page to the browser on the handheld unit.

Figure 1–1 Mobile Point-of-Service Communication Process



Terms

Some terms are used in special ways with handheld devices:

Tap, Select, Click
Tapping and selecting are equivalent to clicking on handheld RF devices, such as the Symbol 8146.

Type, Enter
The terms **Type** or **Enter** mean to input alphanumeric text into the handheld RF device using any means available, such as writing with the stylus or selecting letters or numbers from the on-screen keyboards.

2

Configuration

This chapter covers options for configuring Mobile Point-of-Service normally carried out by an administrator before the system goes into general use.

Setting and Modifying Parameters

Most of the functionality of Mobile Point-of-Service depends directly on Point-of-Service. See the Oracle Retail Point-of-Service Operations Guide for detailed information on configuring the system. Note that the two applications access the same data, from the same store database; however, Mobile Point-of-Service cannot access the database on its own, it must access through the database through Point-of-Service server.

Although it uses the same kinds of parameters as Point-of-Service, Mobile Point-of-Service has its own store.xml and application.xml files for storing XML parameters, so it can use unique parameter values if desired. Because all wireless devices communicate with a single instance of Mobile Point-of-Service, the parameter values set for the Mobile Point-of-Service installation apply to all wireless devices.

Mobile Point-of-Service includes many configurable parameters; these parameters are used to control flow, set minimums and maximums for data, and enable flexibility without recompiling code.

The XML parameters are stored in a set of related parameter XML files. If you change parameter values, then all handheld devices will be affected. This is because there is only one set of parameters files for the Mobile Point-of-Service server.

Each parameter belongs to one and only one group and there is a security role for each group, so parameter access can be restricted to particular groups if desired. To define security roles for users, see “Defining Security with Roles”.

Understanding Parameters XML Tags

Refer to the Oracle Retail Strategic Store Solutions Configuration Guide to understand standard parameter properties.

Parameter File Hierarchy

The Mobile Point-of-Service application gets its parameter values from an interrelated set of XML files. More than one of these files can each contain values for the same parameters; a set of precedence rules determines which parameters actually take effect.

Note: The application.xml file contains all of the parameters and thus represents the default value set. The other files contain subsets of parameters.

Modifying Parameters in Parameter XML Files

Refer to the Oracle Retail Strategic Store Solutions Configuration Guide for information about how to modify parameters in an XML file. Changes made will effect all handheld devices the same way.

Defining Security with Roles

Refer to the procedures in the Point-of-Service Operations Guide for information about how to modify existing roles or add new ones.

RMI Time-out Configuration

To configure remote method invocation (RMI) time-outs refer to the Oracle retail Point-of-Service Operations Guide. The same settings apply to Mobile Point-of-Service except for the name of the file. Use `\OracleRetailStore\mpos\mpos\bin\comm.properties`.

Configuring RMI Hosts and Ports

The RMI host and port configuration are done in the ntier_rmihost.xml file and the web.xml file. In the ntier_rmihost.xml file, both the Mobile Point-of-Service server name and port number and the store server name and port number can be configured. The web.xml file is used to configure the port number that the Web server uses to communicate to the Mobile Point-of-Service server. This port number must match the port number for the Mobile Point-of-Service server defined in the ntier_rmihost.xml file. In the following code samples, values in bold indicate default values set during installation.

Do the following to set the location of the RMI host system for Mobile Point-of-Service:

1. Edit the ntier_rmihost.xml file, found in your `mpos\mpos\lib\com\extendyourstore\unleashed\ config\startup` directory. On the URL tag with the tier UNLEASHED, set the name to the name of your RMI host. Set the port to the port number of the machine running Mobile Point-of-Service. See the following example:

```
<!DOCTYPE CRFHOSTS SYSTEM "classpath://com/extendyourstore/foundation/tour/dtd/
crfhost.dtd" [
]>
<CRFHOSTS transport="RMI">
  <URL name="store_server" port="1300" tier="DATATECHNICIAN"/>
  <URL name="rmi_hostname" port="1097" tier="UNLEASHED"/>
<XMLMANAGER class="XMLManager"
package="com.extendyourstore.foundation.manager.xml"/>
<XMLTECHNICIAN class="XMLTechnician" package="com.extendyourstore.foundation.
manager.xml"/>
</CRFHOSTS>
```

2. Edit the web.xml file found in `360store\360common\jakarta-tomcat-4.1.18\webapps\unleashed\WEB-INF`. For the WebManager servlet tag, set the port param-value to the port number of the machine running the web server. Mobile Point-of-Service must be started at least once for this file to exist. If editing is required, restart Tomcat. See the following example:

```
<servlet>
<servlet-name>WebManager</servlet-name>
<servlet-class>com.extendyourstore.foundation.manager.gui.web.
WebServerCRFServlet</servlet-class>
<init-param>
<param-name>CRFhostname</param-name>
<param-value>localhost</param-value>
</init-param>
<init-param>
<param-name>port</param-name>
<param-value>1097</param-value>
</init-param>
<servlet>
```

Configuring Logging

Mobile Point-of-Service uses the Log4J tool. Configure Log4J by editing `\OracleRetailStore\mpos\mpos\lib\config\log4j.xml`. For more information, see the Apache documentation for Log4J at http://logging.apache.org/log4j.