

**Oracle® Retail Promotion Intelligence and
Promotion Planning and Optimization**

Implementation Guide

Release 13.2

May 2010

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

Primary Author: Judith Meskill

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(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, Oracle Retail Demand Forecasting, Oracle Retail Regular Price Optimization, Oracle Retail Size Profile Optimization, Oracle Retail Replenishment Optimization 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 Mobile Store Inventory Management.

(v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by SAP 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 **DataBeacon™** developed and licensed by Cognos Incorporated of Ottawa, Ontario, Canada, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

Send Us Your Comments	ix
Preface	xi
Audience	xi
Related Documents	xi
Customer Support	xi
Review Patch Documentation	xii
Oracle Retail Documentation on the Oracle Technology Network	xii
Conventions	xii
1 Getting Started	
Introduction	1-1
Documentation	1-2
Implementation Process	1-3
Roles	1-3
PPO Implementation	1-3
PI Implementation	1-3
Process	1-3
Production	1-4
2 Installation	
Introduction	2-1
3 User Management	
Introduction	3-1
About User Roles and User Actions	3-1
About User Management Roles	3-4
User Management Bulk Loader Utility	3-5
Users and Roles	3-5
The xml Files	3-6
Standard Load Prerequisites	3-6
4 PPO UI Configuration	
Introduction	4-1

<configroot>	4-1
PPO Configuration File	4-1
promote.properties	4-2
PPO Configuration File	4-25
Configuring Display Strings	4-25
Configuring Export	4-26
Pull Export Configuration.....	4-27
Integration with Promotion Intelligence	4-27
Auto Authentication Flag	4-28
Report Links Configuration.....	4-28
Display Strings.....	4-28
Debug Messages	4-28

5 Staging and Loading Data

Data Standard Interface	5-1
Standard Load Process	5-1
Environment Customization File.....	5-2
Staging Script: pl_stage_file.sh.....	5-2
Load Script: pl_load_data.sh	5-3
Standard Load Error Handling	5-3
Standard Load Procedures Order	5-4
Standard Load Steps	5-7

6 Database Configuration

Summary Configurations	6-1
CLIENT_HIERARCHY_ACTIONS_TBL	6-4
IR Views	6-4
PR_DB_PARAMS	6-5

7 Managing the Environment

Locating Host Specific Parameters	7-1
Monitoring the PI and PPO Server	7-1
Monitoring Promotion Calc Engine (PCE) Servers	7-2
Monitoring Batch Jobs	7-2
Overview	7-2
Tasks.....	7-2
Steps	7-2
The Job Log Table.....	7-3
Understanding Job Failures.....	7-4
Locating Bad Data.....	7-5
Restarting Jobs.....	7-6
Model Migration	7-6
Parallel Processing	7-7
Manual Process.....	7-7
Running More Than One Process in Operations.....	7-7
Other Maintenance Activities	7-8

Starting and Stopping the Server	7-8
Starting and Stopping the PCE.....	7-8
8 Forecast Accuracy Indicator	
Introduction	8-1
Configuration	8-1
Metrics	8-2
9 Reports	
Introduction	9-1
Available Reports	9-1
Configuration Report Details	9-3
Model Accuracy Scorecard Report	9-3
Report Prompts and Display	9-3
TAE Assessment Report.....	9-3
Report Prompts and Display	9-3
Changing MicroStrategy Summary Levels	9-4
Summary Configurations.....	9-4
MB Counts	9-4
10 Internationalization	
Introduction	10-1
Translation	10-1
Supported Languages	10-2

Send Us Your Comments

Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization Implementation Guide, Release 13.2

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at <http://www.oracle.com>.

Preface

The Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization Implementation Guide provides an overview of the implementation process at a retail customer site.

Audience

This document is intended for administrators of the Promotion Intelligence and Promotion Planning and Optimization application.

Related Documents

For more information, see the following documents in the Oracle Retail Promotion Intelligence and Promotion Planning and Optimization documentation set:

- *Oracle Retail Promotion Intelligence and Promotion Planning and Optimization Release Notes*
- *Oracle Retail Promotion Intelligence and Promotion Planning and Optimization Installation Guide*
- *Oracle Retail Promotion Intelligence and Promotion Planning and Optimization Configuration Guide*
- *Oracle Retail Promotion Intelligence and Promotion Planning and Optimization Operations Guide*
- *Oracle Retail Promotion Intelligence and Promotion Planning and Optimization Data Model*
- *Oracle Retail Promotion Intelligence and Promotion Planning and Optimization Implementation Guide*
- *Oracle Retail Promotion Planning and Optimization User Guide*
- *Oracle Retail Promotion Intelligence and Promotion Planning and Optimization Licensing Information*

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

- <https://metalink.oracle.com>

When contacting Customer Support, please provide the following:

- 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

If you are installing the application for the first time, you install either a base release (for example, 13.0) or a later patch release (for example, 13.1.1). If you are installing a software version other than the base release, be sure to read the documentation for each patch release (since the base release) before you begin installation. Patch documentation can contain critical information related to the base release 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 (with the exception of the Data Model which is only available with the release packaged code):

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.
<i>italic</i>	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.

Getting Started

This chapter introduces the Promotion Intelligence and Promotion Planning and Optimization product and provides an overview of the implementation process. It contains the following sections:

- [“Introduction” on page 1-1](#)
- [“Documentation” on page 1-2](#)
- [“Implementation Process” on page 1-3](#)
- [“Production” on page 1-4](#)

Introduction

Promotion Intelligence (PI) provides a collection of services and stored procedures that are used to assess the historical effectiveness of a set of promotions over a defined time period. The objective is to assess the incremental benefits from promoting items by comparing a promotion period to a baseline period. The resulting reports can help users to evaluate the performance of future promotions.

Promotion Planning and Optimization (PPO) is a web-based application that helps retailers to create and improve promotions. This process includes building offers, designing the ad layout, and producing a forecast to predict how the ad will perform. The user can leverage the information produced by Promotion Intelligence to make the best promotion decisions, using what-if analysis and predictive forecasting.

[Figure 1–1, "Logical System Architecture"](#) shows the relationship between Promotion Intelligence and Promotion Planning and Optimization. The overall process of the application, at a high level, is as follows:

1. Incoming data from the customer is provided to PI as text files and loaded into the database.
2. The Promotion Calc Engine (PCE) process the incoming data and produces forecast parameters.
3. Users, through the PPO UI, enter data into the system and use the application to run What If.
4. Reports are produced against the calculated metrics.
5. Outputs are exported to external systems.

Figure 1–1 Logical System Architecture

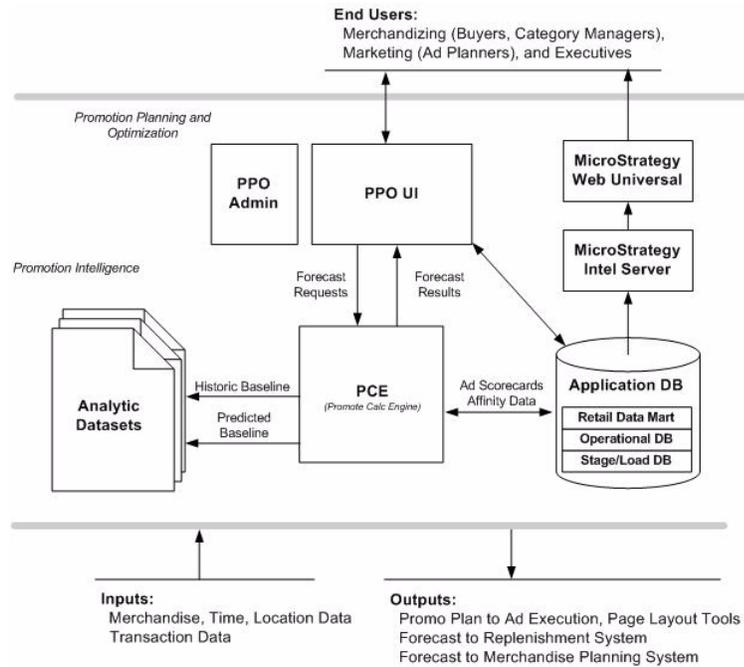
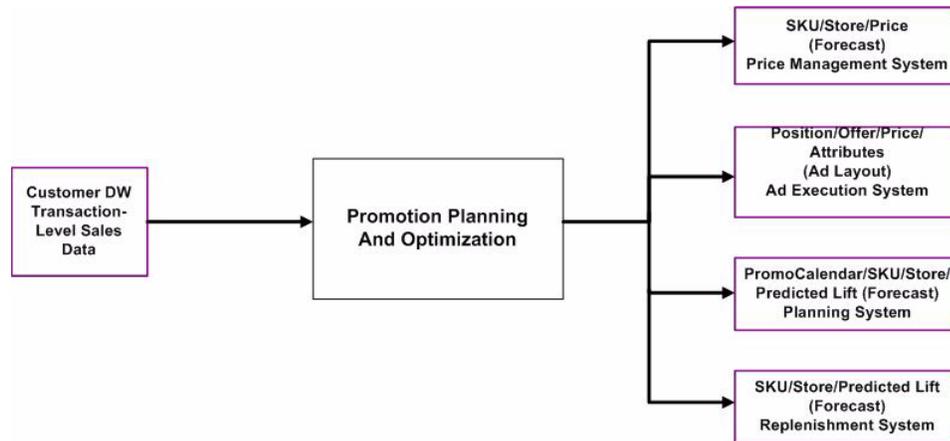


Figure 1–2, "Context Diagram" provides a greater level of detail regarding PPO.

Figure 1–2 Context Diagram



Documentation

The *Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization Implementation Guide* is intended for use with the rest of the documentation set, as listed in the Preface. Here is a description of each of those documents.

The *Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization Installation Guide* provides detailed instructions about planning the installation, setting up the database, setting up the application server, and installing the application.

The *Oracle® Retail Promotion Intelligence User Guide* and the *Promotion Planning and Optimization User Guide* both provide step-by-step procedures for all UI-related tasks.

The *Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization Administration Guide* provides information about managing users and managing stores.

The *Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization Configuration Guide* is a technical reference guide that provides technical details about all application configuration points.

The *Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization Operations Guide* is a technical reference guide that provides technical details about the standard interface, the standard load process, PI tools, and administrative tools.

The *Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization Release Notes* provides a high-level overview of new features and significant issues for the latest release of the application.

Implementation Process

The implementation process creates a client-specific configuration. The configuration is determined by a client's individual business requirements.

Roles

The implementation process typically involves people with the following roles/tasks:

- Database engineer, responsible for data staging and loading
- Analyst, responsible for calculating demand parameters
- Application engineer, responsible for UI configuration
- Production engineer, responsible for the weekly processes

PPO Implementation

The PPO implementation involves configuring the application to match the customer's business needs. This includes managing the formatting and scheduling of the data feeds from the customer, configuring the display of the UI, configuring the RMI server configuration, configuring export, integration with PI, defining and loading the customer's user profiles and store set definitions.

PI Implementation

The PI implementation involves the configuration of the analytical parameters by the analytics team in order to build models and compute ad effectiveness based on the customer's historical data. The process includes:

- Measuring the baseline performance – what would have happened in the absence of promotions - particular to that product but during a normal promotional week (not a dark week)
- Measuring the incremental sales of promoted products (sales above the baseline).
- Performing market basket analysis. The resulting affinity rules show which items sell together and how frequently they sell together.

Process

At a high level, the implementation process consist of these steps:

1. Gather business requirement from the customer. This will include a detailed discussion between the customer and Oracle about how the product works and

what the customer wants to accomplish with the product. The information collected must provide sufficient information for the configuration process to begin. This collection of business requirements and configuration will be an iterative process. Oracle must educate the customer about the data that must be supplied for the configuration. This is a critical discussion and should include information about how the data is supplied and the formats that are required for the data.

2. Configuration design. This step involves determining the data required from the customer for the analytical configuration and the application configuration.
3. The historical data required for the analytical process is obtained from the customer. The analytical calculation are performed iteratively to analyze history, build demand parameters, build lift models, build affinity models, and build predicted models. The resulting values are used for the application configuration.
4. Application configuration. This is a systematic iterative process that requires both historical data and current data. A subset of the configuration occurs and is tested and then other configuration points are added to the configuration and the testing is repeated.
5. User acceptance testing.
6. Training.
7. Transition to a production environment.

Production

Once PI-PPO has been configured and the UAT has been completed, the application can be set up in the customer's production environment. The environment is designed to meet the customer's performance requirements. The nightly and weekly processes are scripted and automated. These processes include a database back up, the loading of the nightly or weekly data, and a status report. Scheduling and process checks are included in the automation. The setting up of the production environment focuses on:

- data transfer mechanism
- data file encryption
- hardware requirements
- operational schedule

Installation

The Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization *Installation Guide* provides instructions for the installation of both Promotion Intelligence and Promotion Planning and Optimization.

Introduction

Refer to this guide for information about the following topics:

- Planning Your Installation
- Setting up the Database
- Setting up the Application Server
- Installation
- Setting Up Single Sign-On
- Installation Order (if installing another Oracle Retail Application)

User Management

This chapter provides technical details about the user management functionality. It contains the following sections:

- “Introduction” on page 3-1
- “About User Roles and User Actions” on page 3-1
- “User Management Bulk Loader Utility” on page 3-5

Introduction

User Management is a utility that lets you create, modify, and remove user accounts from a central location. The User Management utility is installed automatically when you install the application.

Each user who accesses the application must have a user account. Each user account is assigned one or more roles that determine the types of functions the user can perform with the application.

About User Roles and User Actions

Roles are defined by a specific set of user actions. The actions that define each role serve to delimit the activities a user can perform. All actions are self-contained. For example, Create does not imply View. So a role must include all the actions that are necessary for complete functionality.

Note that master data includes the hierarchy information (merchandise, location, and calendar), configuration information (data aggregation levels and other parameters), promotion attributes, store sets information, SKU lists, image information (images and associated mappings for items in the merchandise hierarchy), and dark periods information (time periods that must be excluded from baseline calculations)."

PI-PPO comes with a default set of actions, loaded into ACTION_TBL:

- PROMO_CREATE_CE – add and delete campaigns and events.
- PROMO_EDIT_CE – make changes to campaigns and events.
- PROMO_VIEW_CE – view campaigns and events.
- PROMO_CREATE_MD – add and delete master data.
- PROMO_EDIT_MD – make changes to master data.
- PROMO_VIEW_MD – view master data.
- PROMO_CREATE_PROMO – add and delete promotions.

- PROMO_EDIT_PROMO – make changes to promotions.
- PROMO_VIEW_PROMO – view promotions.
- PROMO_MANAGE_CATEGORY – edit the offers for a promotion.
- PROMO_MANAGE_MERCHANDISE – edit Like Item information.
- PROMO_MANAGE_VEHICLE – edit the definition and design of the promotion vehicle, category assignments, white space allocation, and workflow.
- PROMO_EXPORT_PROMO – provides access to the Export button, which is used to create xml and txt files of promotions. Necessary for access to the Export API functionality.
- PROMO_VIEW_REPORTS – launch the Promotion Intelligence reports.
- PROMO_ADMIN_DOC – only users assigned this action can log into the application when the server is in maintenance mode. Also provides access to the following commands: releaselocks, clearcache, refreshprops, refreshloggin, refreshbundle, refreshconfig, modestage, nodeprod, and modemaint.
- PROMO_APPROVE_OFFER – allows ad planners to approve or deny submitted offers.

PI-PPO comes with a default set of roles, loaded into ROLE_ACTION_TBL:

- PROMO_AD_PLANNER – a member of marketing who is responsible for the entire promotional calendar. This user can create and edit calendar events and create promotions.
- PROMO_BUSINESS_ADMIN – a business user who is responsible for activities such as data maintenance and template management.
- PROMO_CATEGORY_MANAGER – the person directly responsible for one or more categories of merchandise, assigned at a given level in the merchandise hierarchy.
- PROMO_EXEC – an executive who monitors promotion performance across all merchandise categories. Such a user would expect to monitor performance at both a high level and a low level, but would not need to edit or execute promotions.
- PROMO_MERCH_PLANNER – a merchandise planner who executes merchandising plans. Such a user is assigned responsibility at the Chain level.
- PROMO_AGENT – configure users to execute forecast and pre-planned import tasks.
- PROMO_VER_PLANNER – a version planner who executes version plans. Such a user is assigned responsibility at the Chain level.
- PROMO_VER_MANAGER – a version manager who manages version plans. Such a user is assigned responsibility at the Department level.
- PROMO_MERCH_WHATIF – a what-if manager who manages at the Department level.

The following table shows the default assignment of actions to roles in PI-PPO.

Table 3–1 Actions Assigned to Roles

PI-PPO Role	Assigned Actions
PROMO_AD_PLANNER	PROMO_CREATE_CE
	PROMO_EDIT_CE
	PROMO_VIEW_CE
	PROMO_CREATE_MD
	PROMO_EDIT_MD
	PROMO_VIEW_MD
	PROMO_CREATE_PROMO
	PROMO_EDIT_PROMO
	PROMO_VIEW_PROMO
	PROMO_MANAGE_VEHICLE
	PROMO_EXPORT_PROMO
	PROMO_VIEW_REPORTS
	PROMO_APPROVE_OFFER
PROMO_BUSINESS_ADMIN	PROMO_CREATE_MD
	PROMO_EDIT_MD
	PROMO_VIEW_MD
	PROMO_BUSINESS_ADMIN
PROMO_CATEGORY_MANAGER	PROMO_MANAGE_CATEGORY
	PROMO_MANAGE_MERCHANDISE
PROMO_EXEC	PROMO_VIEW_CE
	PROMO_VIEW_MD
	PROMO_VIEW_PROMO
	PROMO_MANAGE_CATEGORY
	PROMO_MANAGE_VEHICLE
	PROMO_EXPORT_PROMO
	PROMO_VIEW_REPORTS
PROMO_MERCH_PLANNER	PROMO_VIEW_CE
	PROMO_VIEW_MD
	PROMO_VIEW_PROMO
	PROMO_EDIT_PROMO
	PROMO_EXPORT_PROMO
	PROMO_VIEW_REPORTS
PROMO_AGENT	PROMO_ADMIN_DOC
	PROMO_VIEW_PROMO
	PROMO_CREATE_PROMO
	PROMO_EDIT_PROMO
	PROMO_VIEW_CE

Table 3–1 (Cont.) Actions Assigned to Roles

PI-PPO Role	Assigned Actions
	PROMO_CREATE_CE
	PROMO_EDIT_CE
	PROMO_VIEW_MD
	PROMO_CREATE_MD
	PROMO_EDIT_MD
	PROMO_MANAGE_MERCHANDISE
	PROMO_MANAGE_CATEGORY
	PROMO_MANAGE_VEHICLE
	PROMO_EXPORT_PROMO
	PROMO_VIEW_REPORTS
PROMO_VER_PLANNER	PROMO_VIEW_CE
	PROMO_VIEW_MD
	PROMO_VIEW_PROMO
	PROMO_EDIT_PROMO
	PROMO_CREATE_PROMO
	PROMO_EXPORT_PROMO
	PROMO_VIEW_REPORTS
PROMO_VER_MANAGER	PROMO_MANAGE_VEHICLE
PROMO_MERCH_WHATIF	PROMO_VIEW_CE
	PROMO_VIEW_MD
	PROMO_VIEW_PROMO
	PROMO_EDIT_PROMO
	PROMO_EXPORT_PROMO
	PROMO_VIEW_REPORTS

Default actions cannot be deleted.

Roles are assigned to users with restrictions that are defined at or above a specific node of the merchandise hierarchy and the location hierarchy. The scope of actions can be across the merchandise and location hierarchies.

The sample file, "Role Assignment Sample xml File" provides an illustration of defining the scope.

About User Management Roles

User accounts with user management roles have access to features such as creating users, assigning roles, removing user accounts, resetting passwords.

When a user with a user management role logs on, a link to the User Management utility appears on the Main Menu.

The following list describes the default User Management roles:

- `UM_READ_ONLY_ADMIN` – This role allows read-only access to the User Management utility. This role has privileges to view the list of users and their roles and hierarchy levels, but not to create new user accounts or modify or inactivate existing ones.
- `UM_ROLE_ASSIGN_ADMIN` – This role allows assigning new roles (and related hierarchy levels) to existing user accounts, but it does not allow the creation of new user accounts.
- `UM_USER_ADMIN` – This role allows creating new user accounts, but it does not allow the assignment of roles to the new accounts.

User Management Bulk Loader Utility

To create user accounts for a group of users all at one time, you can use the User Management bulk loader utility.

Prior to running the User Management bulk loader utility, you must:

- Set the `jndi.properties`. The `jndi.properties` file, which is located in `<installed>/modules/tools/conf/jndi.properties`, specifies the initial context factory and the url where the JNDI lookups are carried out.

For WebLogic, typical values are:

```
java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory
java.naming.provider.url=t3://localhost:7001
```

- Make sure that `usermanagement.ear`, `suiteproperties.ear`, and `common4p.ear` are deployed on the running application server.

Users and Roles

You need to create and validate (using a tool like XML Spy) three xml files containing entries for Users, Roles, and Role Assignments.

Note that the actions associated with roles must be created, using `brmadmin.sh` in order for the roles to be successfully created.

- The user file contains user names. All user names must be unique. The schema includes a flag that indicates whether or not the password should be hashed.
- The Roles file contains the possible roles that can be assigned. All role keys must be unique. The action key attributes must be loaded into the database before the bulk loader utility can be used. All elements and attributes must be lower case.
- The Role Assignment file contains user names and the role or roles associated with the user name. The user names must be loaded into the database before this file can be processed by the bulk loader utility. All elements and attributes must be lower case. The merchandise ID and the Location ID are provided by a pipe-delimited string of `CLIENT_LOAD_ID`, as found in the `MERCHANDISE_HIERARCHY_TBL` or `LOCATION_HIERARCHY_TBL`. For example, to assign a user to a certain department of merchandise:

CHAIN COMPANY DIVISION DEPARTMENT merchandise attribute in .xml

```
-----
0 1 123 8765 1 | 123 | 8765
0 1 22 789 1 | 22 | 789
```

The information in the three files is loaded into database tables by the bulk loader. (Users and Role Assignments can be added or modified via the application UI. Roles can only be added or modified via the bulkloader.)

The xml Files

The xml schemas and samples of the three required xml files can be found in <installed>/modules/tools/conf.

Table 3–2 User Management xml Files

Schema	Sample	Database Table
user-set.xsd	test_user_set.xml	USERS_TBL
role-set.xsd	test_role_set.xml	ROLES_TBL
role-assignment-set.xsd	test_assignment_set.xml	USER_RESOURCE_ROLE_TBL

Standard Load Prerequisites

Before you run the bulk loader, you must have run the standard load so that the merchandise hierarchy table (ASH_MH_TBL) and the location hierarchy table (ASH_LH_TBL) have been populated. (For more information on the standard load, see the *Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization Operations Guide*).

PPO UI Configuration

This chapter provides an overview of the configuration of the Promotion Planning and Optimization UI. For more details, consult the *Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization Configuration Guide*.

This chapter contains the following sections:

- “Introduction” on page 4-1
- “<configroot>” on page 4-1
- “PPO Configuration File” on page 4-25
- “Configuring Display Strings” on page 4-25
- “Configuring Export” on page 4-26
- “Integration with Promotion Intelligence” on page 4-27
- “Debug Messages” on page 4-28

Introduction

The PPO product includes a configurable Graphical User Interface (GUI). Several configuration points can be used to modify GUI behavior.

<configroot>

<configroot> is the entry point directory that is used by the application to look up all the configuration files. This value has to be set at the application server (OAS) level. Refer to the *Promotion Installation Guide* for instructions on how to set it. It usually points to <install-dir>/config.

PPO Configuration File

GUI-wide properties are set in the `promote.properties` file, which is located in <configroot>/promote. This file is pre-populated during the installation process with installation-specific values.

Other properties relate to OAS 10.1.3.1 and OAS 10.1.3.3 and contain the prefix `oas`. For example, `oas.java.naming.provider.url` has to be set to the correct (opmn or non-opmn) url, such as `ormi://host:port`.

The same is true for the configuration of the User Management application. Its properties are stored in <configroot>/usermanagement/usermanagement.properties.

To specify the time interval at which PI-PPO Planning updates the server session on browser-only user activity (that is, when a user click does not result in a server call), set `promotekeepalive.interval` to the desired interval (in seconds).

promote.properties

Note that all internationalization configuration settings are contained in `promote.properties` and all translated text for display in the UI are contained in `promoteResources.properties`.

Here are details and the default settings for the `promote.properties` file:

The system configuration properties are used to define necessary system configuration parameters.

Table 4–1 System Configuration Properties

Property	Description
<code>promote.config.file=promote-config.xml</code>	Identifies the location of file used for integration with Promotion Intelligence (PI).
<code>promote.saxparser.classname=org.apache.xerces.parsers.SAXParser</code>	Name of xml parser.
<code>promote.server.mode=prod</code>	<p>The mode values are:</p> <p><code>dev</code> = an optional mode used only by developers for debugging. It enables HTTP get request, supports addition debug URL commands, and disables master data caching.</p> <p><code>sqa</code> = reserved for future use by QA.</p> <p><code>impl</code> = reserved for future use by implementation team for debugging.</p> <p><code>stage</code> = reserved for future use for staging system features.</p> <p><code>prod</code> = the default. It enables the server to maintain a cache of master data (such as merchandise/location hierarchy data).</p> <p><code>maint</code> = allows a production to temporarily disable user logins but allow system users to perform various maintenance activities.</p>

The export properties are used for mapping between the UI and the export file. These entries can be added to or changed. Export keys used in the UI take the form of `promte.export.template + template name + output file extension + short name of the exported domain class`.

Table 4–2 Export Properties

Property	Description
promote.export.path=%{installdir}%/config/promote	Identifies the directory for the templates.
The export keys used in the UI take the form promote.export.template + output file extension + short name of the exported domain class, to lower class.	
promote.export.template.AllOffersForecast.csv.promotion=AllOffersForecastExportTemplate.xslt	Mapping for AllOffersForecast (csv).
promote.export.template.AllOffers.csv.promotion=AllOffersDetailExportTemplate.xslt	Mapping for AllOffers (csv).
promote.export.template.Layout.xml.promotion=LayoutXMLExportTemplate.xslt	Mapping for Layout (xml).
promote.export.template.Layout.html.promotion=LayoutExportTemplate.xslt	Mapping for Layout (html).
promote.export.template.xml.promotion=XmlExportTemplate.xslt	Mapping for xml.
promote.export.template.AllSKU.csv.promotionoffer=AllSKUExportTemplate.xslt	Mapping for AllSKU (csv).
promote.export.template.Versions.csv.promotionoffer=VersionsExportTemplste.xslt	Mapping for Versions (csv).
promote.export.template.txt.promotionoffer=TxtExportTemplate.xslt	Mapping for Offer (txt).
promote.export.template.xml.promotionoffer=XmlOfferExportTemplate.xslt	Mapping for Offer (xml).
The xml export keys take the form promote.export.template.xml + short name of exported domain class, to lower case.	
promote.export.template.xml.promotionofferpositionsummary=XmlCmdlineExportTemplate.xslt	Mapping for promotionofferposition (xml).
promote.export.template.xml.promotionofferfullsummary=XmlCmdlineExportTemplate.xslt	Mapping for promotionofferfullsummary (xml).
The txt export keys take the form promote.export.template.txt + short name of the exported domain class, to lower case.	
promote.export.template.txt.promotionofferpositionsummary=PromoOfferPosSumTxtTmpl.xslt	Mapping for promotionofferpositionsummary (txt).
promote.export.template.txt.promotionofferfullsummary=PromoOfferSumTxtTmpl.xslt	Mapping for promotionofferfullsummary (txt).
The filter keys used in the UI take the form promote.filter.columns + short name of the exported domain class, to lower case. The values are comma-separated lists of column names.	
promote.filter.columns.promotionoffercategoryattribute=none,vendorID,retail,cost,purchaseType,group	Mapping for promotionoffercategoryattribute (csv).
promote.filter.columns.promotionofferriterionattributetranslator=none,vendorID,retail,cost,purchaseType,group	Mapping for promotionofferriterionattribute (csv).
The xslt templates for computing promo properties. The keys used in the UI include promote + property +template +<short name of the domain class>	
promo.eventexternalname.template.promotion=DefaultPromoEventExtName.xslt	Mapping for promotion (xslt).
promo.eventexternalname.template.promotionoffer=DefaultPromoOfferEventExtName.xslt	Mapping for promotionoffer (xslt).

The agent properties are used to configure the scheduling and performance of agents. The agents include ones for the pre-planned promotion loader, forecasts, weekly forecasts, email properties, audit trail properties, and formatting for dates and numbers. Ensure that the schedules for the forecast and the weekly forecast do not clash. The agent configuration consists of name of task class + command to servlet. The same tasks can be used by multiple agents.

Table 4-3 Agent Properties

Property	Description
promote.agent.delay=2	Delay schedule.
promote.agent.url=http://%{suite.host}:%{suite.port}/promote/export.do	Agent url.
promote.agent.run.policy=fixed_delay	Policy for submitting agent requests. Values allowed: fixed_rate and fixed_delay (the default). This is a global setting for all agents and determines the type of delay between agent requests. See also, for example, promote.agent.preplannedpromo.delay.
Each agent requires a user password and the user must have PromoAgent permission. There should be as many as the maximum number of threads for all the agents that have a common schedule. Must be a unique set per node in a cluster. For example, for one thread for preplannedpromo + 2 threads per forecast, you will need at least three users and corresponding passwords. These passwords must be hashed using suite_lib.jar.	
promote.agent.user.1=sysid0	Agent 1 id.
promote.agent.password.1=	Agent 1 password.
promote.agent.user.2=sysid1	Agent 2 id.
promote.agent.password.2=	Agent 2 password.
promote.agent.user.3=sysid2	Agent 3 id.
promote.agent.password.3=	Agent 3 password.
promote.agent.user.4=sysid3	Agent 4 id.
promote.agent.password.4=	Agent 4 password.
promote.agent.user.5=sysid4	Agent 5 id.
promote.agent.password.5=	Agent 5 password.
promote.agent.user.6=sysid5	Agent 6 id.
promote.agent.password.6=	Agent 6 password.
promote.agent.user.7=sysid6	Agent 7 id.
promote.agent.password.7=	Agent 7 password.
promote.agent.user.8=sysid7	Agent 8 id.
promote.agent.password.8=	Agent 8 password.
promote.agent.user.9=sysid8	Agent 9 id.
promote.agent.password.9=	Agent 9 password.
promote.agent.user.10=sysid9	Agent 10 id.
promote.agent.password.10=	Agent 10 password.
promote.translate.storeset=Default	Translation configuration for storeset.
promote.translate.storesubset=Central	Translation configuration for store subset.
promote.translate.forecast=false	Translation configuration for forecast.

Table 4–3 (Cont.) Agent Properties

Property	Description
promote.agent.preplannedpromo.command=runPreplannedPromoLoad	Calls the pre-planned promotion loader.
promote.agent.preplannedpromo.enabled=false	Defines whether or not the x is enabled. True = enabled; False = not enabled. The default is false.
promote.agent.preplannedpromo.runonce=false	Defines how often to run the agent for a given period. False (default) = continually. True = once per period.
promote.agent.preplannedpromo.runweeks=1-52	The weeks assigned for the agent to run. Use either a comma-separated list of numbers (1 through 52 in ascending order) for individual weeks or two numbers separated by a hyphen for a range of weeks. (For example, 1,5,9,13... would define a one-week-a-month schedule.) The default is all weeks. The schedules for different instances of the same task should not overlap.
promote.agent.preplannedpromo.rundays=Mon,Tue,Wed,Thu,Fri,Sat,Sun	The day or days of the week assigned for the agent to run. Use a comma-separated list of days of the week (either Mon/Tue/Wed/Thu/Fri/Sat/Sun or the full name of the day). Cannot be localized. The default is all days of the week.
promote.agent.preplannedpromo.start=19:30	The start time for the agent schedule in a 24-hour format of hh:mm. Note that an end date is also required, or the agent will run continuously.
promote.agent.preplannedpromo.end=23:30	The end time for the agent schedule in a 24-hour format of hh:mm.
promote.agent.preplannedpromo.delay= 60	Defines the length of the delay in seconds. See also promote.agent.run.policy.
promote.agent.preplannedpromote.num_threads = 1	Number of threads for preplanned promo.
promote.agent.preplannedpromo.max_num_threads=1	Recommended value is 1.
promote.agent.promoforecast.command=runPromotionForecastor	Calls the loader.
promote.agent.promoforcast.enabled=false	Defines whether or not the agent is enabled. True = enabled; False = not enabled. The default is false.
promote.agent.promoforest.runonce=false	Defines how often to run the agent for a given period. False (default) = continually. True = once per period.
promote.agent.promoforecast=runweeks=1-52	The weeks assigned for the agent to run. Use either a comma-separated list of numbers (1 through 52 in ascending order) for individual weeks or two numbers separated by a hyphen for a range of weeks. (For example, 1,5,9,13... would define a one-week-a-month schedule.) The default is all weeks. The schedules for different instances of the same task should not overlap.
promo.agent.promoforecast.rundays=Mon,Tue,Wed,Thu,Fri,Sat,Sun	The day or days of the week assigned for the agent to run. Use a comma-separated list of days of the week (either Mon/Tue/Wed/Thu/Fri/Sat/Sun or the full name of the day). Cannot be localized. The default is all days of the week.

Table 4–3 (Cont.) Agent Properties

Property	Description
promote.agent.promoforecast.start=23:30	The start time for the agent schedule in a 24-hour format of hh:mm. Note that an end date is also required, or the agent will run continuously.
promo.agent.promoforecast.end=6:30	The end time for the agent schedule in a 24-hour format of hh:mm.
promo.agent.promoforecast.delay=60	Defines the length of the delay in seconds. See also promote.agent.run.policy.
promote.agent.promoforecast.num_threads=1	Number of threads for promo forecast.
promote.agent.promoforecast.max_num_threads=1	Recommended value is 1.
promote.agent.promoforecast.refresh=true	Indicates whether refresh occurs.
promote.agent.promoforecast.forecast=true	Indicates whether forecast occurs.
promote.agent.forecast.command=runForecast	Forecast task
promote.agent.forecast.enabled=false	Defines whether or not the agent is enabled. True = enabled; False = not enabled. The default is false.
promote.agent.forecast.runonce=false	Defines how often to run the agent for a given period. False (default) = continually. True = once per period.
promote.agent.forecast.runweeks=1-26,27-52	The weeks assigned for the agent to run. Use either a comma-separated list of numbers (1 through 52) for individual weeks or two numbers separated by a hyphen for a range of weeks. (For example, 1,5,9,13... would define a one-week-a-month schedule.) The default is all weeks.
promote.agent.forecast.rundays=Mon,Tue,Wed,Thu,Fri	The day or days of the week assigned for the agent to run. Use a comma-separated list of days of the week (either Mon/Tue/Wed/Thu/Fri/Sat/Sun or the full name of the day). Cannot be localized. The default is all days of the week.
promote.agent.forecast.start=23:30	The start time for the agent schedule in a 24-hour format of hh:mm. Note that an end date is also required, or the agent will run continuously.
promote.agent.forecast.end=6:30	The end time for the agent schedule in a 24-hour format of hh:mm.
promote.agent.forecast.delay=60	Defines the length of the delay in seconds. See promote.agent.run.policy.
promote.agent.forecast.num_threads=1	Number of threads for forecast.
promote.agent.forecast.max_num_threads=1	Recommended value is 1.
promote.agent.forecast.fromdate=01/01/2007 00:00:00	Agents search for promotions to forecast that begin on a date that is greater than this date. Format is based on promote.datetime.us.timestamp.
promote.agent.forecast.todate=01/01/2007 00:00:00	Agents search for promotions to forecast that end on a date that is less than this date. The todate must be greater than the fromdate. Format is based on promote.datetime.us.timestamp. If omitted, defaults to sysdate.

Table 4–3 (Cont.) Agent Properties

Property	Description
promote.agent.forecast.uptodate=03/31/2007 00:00:00	The uptodate must be greater than the todate. If this value is omitted, all future promotions will be forecast. Format is based on promote.datetime.us.timestamp.
promote.agent.forecast.type=	Indicates which type. 0 =promotion created in UI. 4 = historical promotion. 5 = pre-planned promotion received from client. If no value - all types will be included.
promote.agent.forecast.phase=	Indicates which workflow. If no value - all phases will be included.
promote.agent.forecast.status=	Values include forecast (to do), current (in process), outofdate (re-forecast), failed, and expired. An empty or non-force value produces forecast for only Out-of-date and No forecast.
promote.agent.forecast.groupby=false	Flag that indicates whether to not to do multiple promotions. The default - true - groups promotions.
promote.agent.forecast.forecast=trur	False = in order that refresh can be done separately, do not forecast. True (default) = forecast.
promote.agent.forecast.force=false	If true, forecast even if the offer status is CURRENT/PARTIAL after the offer is opened and the status is recomputed. If false (the default), do not forecast CURRENT/PARTIAL offers (as of the last db status).
promote.agent.forecast.refresh=false	Flag that indicates whether SKUS are refreshed before the forecast. If set to true, the SKUs are refreshed, which adds time to the forecast.
promote.agent.forecastweekly.command=runForecast	Weekly forecast task
promote.agent.forecastweekly.enabled=false	Defines whether or not the agent is enabled. True = enabled; False = not enabled. The default is false.
promote.agent.forcastweekly.runonce=false	Defines how often to run the agent for a given period. False (default) = continually. True = once per period.
promote.agent.forecastweekly.runweeks=1-26,27-52	The weeks assigned for the agent to run. Use either a comma-separated list of numbers (1 through 52) for individual weeks or two numbers separated by a hyphen for a range of weeks. (For example, 1,5,9,13... would define a one-week-a-month schedule.) The default is all weeks.
promote.agent.forecastweekly.rundays=Sun	The day or days of the week assigned for the agent to run. Use a comma-separated list of days of the week (either Mon/Tue/Wed/Thu/Fri/Sat/Sun or the full name of the day). Cannot be localized. The default is all days of the week.
promote.agent.forecastweekly.start=23:35	The start time for the agent schedule in a 24-hour format of hh:mm. Note that an end date is also required, or the agent will run continuously.
promote.agent.forecastweekly.end=06:30	The end time for the agent schedule in a 24-hour format of hh:mm.

Table 4–3 (Cont.) Agent Properties

Property	Description
promote.agent.forecastweekly.delay=60	Defines the length of the delay in seconds. See promote.agent.run.policy.
promote.agent.forecastweekly.num_threads=1	Number of threads for weekly forecast.
promote.agent.forecastweekly.max_num_threads=1	Recommended value is 1.
promote.agent.forecastweekly.fromdate=01/01/2007 00:00:00	Agents search for promotions to forecast that begin on a date that is greater than this date. Format is based on promote.datetime.us.timestamp.
promote.agent.forecastweekly.todate=01/01/2007 00:00:00	Agents search for promotions to forecast that end on a date that is less than this date. The todate must be greater than the fromdate. Format is based on promote.datetime.us.timestamp. If omitted, defaults to sysdate.
promote.agent.forecastweekly.uptodate=03/31/2007 00:00:00	The uptodate must be greater than the todate. If this value is omitted, all future promotions will be forecast. Format is based on promote.datetime.us.timestamp.
promote.agent.forecastweekly.type=	Indicates which type. 0 = promotion created in UI. 4 = historical promotion. 5 = pre-planned promotion received from client. If no value - all types will be included.
promote.agent.forecastweekly.phase=	Indicates which workflow. If no value - all phases will be included.
promote.agent.forecastweekly.status	Values include forecast (to do), current (in process), outofdate (re-forecast), failed, and expired. An empty or non-force value produces forecast for only Out-of-date and No forecast.
promote.agent.forecastweekly.groupby=false	Flag that indicates whether to not to do multiple promotions. The default - true - groups promotions.
promote.agent.forecastweekly.forecast=true	Flag that indicates whether or not to forecast so that a refresh can occur. If value is true (default) the forecast is done.
promote.agent.forecastweekly.force=false	If true, forecast even if the offer status is CURRENT/PARTIAL after the offer is opened and the status is recomputed. If false (the default), do not forecast CURRENT/PARTIAL offers (as of the last db status).
promote.agent.forecastweekly.refresh=false	Flag that indicates whether SKUs are refreshed before the forecast. If set to true, the SKUs are refreshed, which adds time to the forecast.

The email properties are used to configure system email.

Table 4–4 Email Properties

Property	Description
promote.mail.smtp.host	The IP address of the SMTP host.

Table 4–4 (Cont.) Email Properties

Property	Description
promote.mail.smtp.port	The number of the SMTP port.
promote.mail.username	The username to use when connecting to the email server.
promote.mail.password	The password to use when connecting to the email server.

The following properties are used to define the error reporting feature in the application:

Table 4–5 Error Reporting Properties

Property	Description
The following parameters determine the information that is sent in a client error report E-mail.	
promote.client.mail.includeFlashVersion=true	Flag to include the Flash version in the E-mail.
promote.client.mail.includeOS=true	Flag to include the operating system version in the E-mail.
promote.client.mail.includeResolution=true	Flag to include the screen resolution in the E-mail.
promote.client.mail.includeMemory=true	Flag to include the system memory in the E-mail.
promote.client.mail.includeURL=true	Flag to include the application URL in the E-mail.
promote.client.mail.includeBrowserDetails=true	Flag to include the Web browser information in the E-mail.
promote.client.mail.includeProductVersions=true	Flag to include the application versions in the E-mail.
<i>The following parameters determine the information that is logged when an unexpected error or configuration error occurs in the client.</i>	
promote.client.error.includeFlashVersion=true	Flag to include the Flash version in the log.
promote.client.error.includeOS=true	Flag to include the operating system version in the log.
promote.client.error.includeResolution=false	Flag to include the screen resolution in the log.
promote.client.error.includeMemory=false	Flag to include the memory information in the log.
promote.client.error.includeURL=true	Flag to include the application URL in the log.
promote.client.error.includeBrowserDetails=true	Flag to include the Web browser information in the log.
promote.client.error.includeProductVersions=false	Flag to include the application versions in the log.

The audit trail properties are used to activate and restrict auditing.

Table 4–6 Audit Trail Properties

Property	Description
promote.audit.promotion=false	Flag to activate auditing of promotion. Default is true.
promote.audit.promotionoffer=false	Flag to activate auditing of promotion offer. Default is true.
promote.audit.promotionoffer.userdefinedfields=false	Flag to activate auditing of user defined fields for promotion offer. Default is false.

Table 4–6 (Cont.) Audit Trail Properties

Property	Description
promote.audit.promotionvehicle=false	Flag to activate auditing of the promotion vehicle. Default is false.
promote.audit.promotionvehiclepage=false	Flag to activate auditing of promotion vehicle page. Default is false.
promote.audit.forecast=false	Flag to activate auditing of forecast. Default is false.
promote.audit.forecast_value=units	Possible values of forecast are units, sales, and margin.
promote.audit.same.user=false	Audits all users' changes except for changes made by the same user to the same object in succession.
promote.audit.system.user=false	Excludes changes triggered by system and batch processes users.

The miscellaneous properties are used to identify system URLs and engine passwords.

Table 4–7 Miscellaneous Properties

Property	Description
promote.engine.url=rmi://%{KDE_RMI_SERVER_ADDRESS}:%{KDE_RMI_SERVER_PORT}%/ItemPredictorFactory	Engine property.
promote.engine.user=root	Engine root password.
promote.engine.password=	This password must be hashed using suite_lib.jar. java com.profitlogic.common.security.util.PasswordUtility clearPassWord
promote.imageserver.baseurl=http://%{suite.host}:%{suite.port}%/iserver/images/mh	Image server connection information.
promote.help.url=http://%{suite.host}:%{suite.port}%/ppohelp/help	Help URL

The locale properties specify country and language. For each country and language specified, the corresponding translated promoteResources.properties for that language are used for display in the UI.

Table 4–8 Locale Properties

Property	Description
promote.locale.country=US	Specifies the locale being used.
promote.locale.language=en	Specifies the language for the locale.

The date properties define the formatting for dates.

Table 4–9 Date Properties

Property	Description
promote.dateformat.lenient=false promote.datetime.dateformatpolicy=configured	<p>Values are:</p> <p>metadata = the format coded by the server developer for the class (currently there are none), reserved for future use.</p> <p>bean = the format coded by the server developer for the instance (currently there are none), reserved for future use.</p> <p>request = the default format of the java virtual machine for the locale specified in the browser.</p> <p>system = the default format of the java virtual machine for the locale specified in the server.</p> <p>configured (the default) = uses the format configured in promote.properties.</p>
promote.datetime.???date promote.datetime.???time promote.datetime.???timestamp	Specified value for formatting is substituted for date, time, and timestamp keys.
promote.datetime.sysdateformat=us	Defines the country for the formatting.
promote.datetime.eu.date=dd/MM/yyyy	Defines the date format for the eu country.
promote.datetime.us.date=MM/dd/yyyy	Defines the date format for the us.
promote.datetime.iso.date=yyyy-MM-dd	Defines the iso date format.
promote.datetime.us.time=HH:mm:ss	Defines the time format for the us.
promote.datetime.iso.time=HH:mm:ss	Defines the iso time format.
promote.datetime.eu.time=HH:mm:ss	Defines the time format for the eu country.
promote.datetime.us.timestamp=MM/dd/yyyy HH:mm:ss	If the date and time formats are known, but the timestamp is missing, it will be created by concatenating (separated by a space) the date and time formats.
promote.datetime.iso.timestamp=yyyy-MM-dd HH:mm:ss	If the date and time formats are known, but the timestamp is missing, it will be created by concatenating (separated by a space) the date and time formats.
promote.datetime.eu.timestamp=dd/MM/yyyy HH:mm:ss	If the date and time formats are known, but the timestamp is missing, it will be created by concatenating (separated by a space) the date and time formats.
label.header.format.date=M/d/yyyy	Defines the calendar header.
label.header.format.month=MMMM yyyy	Defines the calendar header.
label.header.format=yyyy	Defines the calendar header.

The number properties must be valid Java number formats.

Table 4–10 Number Properties

Property	Description
promote.format.int=#,##0	Integer format.
promote.format.decimal=#,##0,###	Decimal format.
promote.format.percent=#,##0,##%	Percent format.
promote.format.currency=\u00A4#,##0.00	\u00A4 is the unicode general currency symbol, which java localizes to the currency symbol.
promote.metric.int=#,##0.0	Metric integer format
promote.metric.decimal=#,##0,###	Metric decimal format.
promote.metric.percent=#,##0,##%	Metric percent format.
promote.metric.currency=\u00A4#,##0	\u00A4 is the unicode general currency symbol, which java localizes to the currency symbol.

The data properties must be valid Java integers in the valid range 200-5000 with a default if unspecified of 1000. If the value exceeds 5000, the limit will be 5000. If the value is below 200, the limit will be 200.

Table 4–11 Data Properties

Property	Description
promote.offer.max.items=25000	Maximum number of items in an offer.
promote.offer.max.itemlocations=75000	Maximum number of item locations in an offer.
promote.data.maxrows=1000	Maximum number of entries (rows of data) that display in the application.
promote.data.caselessSearch=true	Ignore the case in a search operation.
promote.data.wildcardSearch=true	Use wild cards in a search operation. If set to True and multiple words are placed in the search string, the wild cards are placed around all words within the search string.
promote.data.likeSearch=true	Use the "like" option in a search operation. If set to True, the search performs an SQL LIKE comparison in which all or some of the words are searched for, instead of a SQL = comparison search, in which only exact matches are returned. If set to False, the wild card property does not work.
promote.data.caselessSort=false	Sets the data to sort without considering the case of the text.
promote.data.nullsHighSort=true	Indicates whether or not nulls are sorted high.
promote.data.sortDepth=2	Number of columns to preserve sorting when user clicks on column heading.
promote.pagesize.lookahead=1	Number of pages of data to cache to client.
promote.offer.criterion.filternum=5	Number of offer category criteria filters.
promote.unpositioned.offer.rollup=false	Indicates whether or not unpositioned offers metrics roll up into promotion/category.
promote.versioned.position.rollup=true	Indicates whether or not versioned positions metrics roll up into the base page metric.
promote.persist.cat.rollup=false	Indicates whether or not category metric roll ups are persisted.

Table 4–11 (Cont.) Data Properties

Property	Description
promote.persist.promo.rollup=false	Indicates whether or not promotion metric roll ups are persisted.
promo.persist.page.rollup=false	Indicates whether or not page metric roll ups are persisted.
promote.persist.vehicle.rollup=false	Indicates whether or not vehicle metric roll ups are persisted.
promote.persist.position.rollup=false	Indicates whether or not position metric roll ups are persisted.
promote.persist.item.locations=true	Indicates whether or not promotion offer item location records are persisted.
promote.promotion.items.unique=false	Use this parameter (set to false) to specify that the promotion can include the same item more than once.
promote.positioned.offers.unique=false	Use this parameter (set to false) to specify that the offers can be positioned more than once.
promote.categories.min.lcd=false	Indicates whether or not promotion categories are enforced to equal the min_lcd setting in ash_cp_tbl.
promote.multiple.vehicles=true	<p>Use this parameter to specify that the promotions and promotion templates can have more than one vehicle (set to true). When set to false, the following screen elements will be disabled and appear greyed out:</p> <ul style="list-style-type: none"> ■ For promotions: <ul style="list-style-type: none"> – Add and Delete buttons on the Vehicles tab. – Vehicle drop-down list on the Vehicle Design tab. ■ For promotion templates: <ul style="list-style-type: none"> – Add and Delete buttons on the Vehicles tab.
promote.vehicle.versions=true	Use this parameter to specify that the promotion vehicles and vehicle templates can have versions defined (set to true). When set to false, the Versions tab will be disabled and may not appear on screen for promotion vehicles and vehicle templates.
promote.vehicle.versions.demanddrivers.sync=true	<p>Use this parameter to ensure that any changes to the demand driver value for a positioned offer version get reflected to the corresponding demand driver values for the other versions in the same vehicle.</p> <p>When set to true, changing the vehicle's demand driver value will be cascaded to the corresponding demand drivers for the other versions in the same vehicle.</p> <p>When set to false, the previous functionality will be retained where changing the version's demand driver value did not impact the other versions. Users will require to manually update the demand driver values.</p> <p>Note: This configuration applies only for the positioned offer versions and the What-If scenarios.</p>

Table 4–11 (Cont.) Data Properties

Property	Description
promote.versioned.pages=true	Use this parameter to specify that the promotion vehicle pages and vehicle template pages can be defined for different versions (set to true). When set to false, the Versions drop-down list will be disabled and appear greyed out on the Vehicle Design tab in the Promotion and Vehicle Template window. Note: When the <i>promote.positioned.offers.unique</i> is set to true and <i>promote.versioned.pages</i> is set to false, the version check boxes for the positioned offers will not be disabled. The version check boxes appear at the following locations in the application: <ul style="list-style-type: none"> ■ Version Details panel from the Version View on the Offer Definition tab. ■ Modify Versions panel for the Current Offer in the Performance & What-If tab.
promote.versioned.positions=true	Indicates whether or not positions are versioned.
promote.version.date.overrides=true	Use this parameter to enable the users to override the start and end dates for a version.
promote.vehicle.date.overrides=true	Use this parameter to enable the users to override the start and end dates for a vehicle.
promote.offer.date.overrides=true	Use this parameter to enable the users to override the start and end dates for an offer.
promote.version.grouping=true	Use this parameter to enable version grouping in the application.
promote.offer.version.grouping=false	Use this parameter to enable version grouping for offers. When enabled, Offers list screen, Offer details and Performance & What-If tabs will show the version groups associated with the offer. If set to false, the offers will show the versions within the associated version groups.
promote.category.manager.lock.pages=true	Use this parameter to enable users with the access privileges of a Category Manager to automatically lock the entire page when they open a page for editing (in place of locking each individual position).
promote.valueindex.affinity.editable=true	Indicates whether or not affinity value index ratios are editable.
promote.valueindex.forecast.editable=true	Indicates whether or not forecast value index ratios are editable.
promote.valueindex.editable=true	Indicates whether or not value index ratios are editable or not. This requires either <i>promote.valueindex.affinity.editable</i> or <i>promote.valueindex.forecast.editable</i> to be true as well.
## Data load properties	
promote.load.offer.versions=true	Indicates whether or not version records are loaded when the promotion is loaded.
promote.load.storesets.startup=true	Indicates whether or not storesets are deep loaded on start-up. If true, this also requires <i>promote.cache.feed.storeset</i> to be true.
promote.load.skulists.startup=true	Indicates whether or not SKU lists are deep loaded on start-up. If true, this also requires <i>promote.cache.feed.skulist</i> to be true.
promote.load.hierarchy.startup=false	Indicates whether or not location/merchandise hierarchies are deep loaded on start-up. If true, this also requires <i>promote.cache.feed.location</i> or <i>promote.cache.feed.merchandise</i> to be true as appropriate.
promote.cache.feed.location=true	Indicates whether or not the location feed should be cached.
promote.cache.feed.merchandise=true	Indicates whether or not the hierarchy feed should be cached.

Table 4–11 (Cont.) Data Properties

Property	Description
promote.cache.feed.storeset=true	Indicates whether or not the storeset feed should be cached.
promote.cache.feed.skulist=true	Indicates whether or not the skulist feed should be cached.
promote.cache.feed.userentry=true	Indicates whether or not the user feed should be cached.
promote.cache.feed.image=true	Indicates whether or not the image feed should be cached.
promote.roi.includes.affinity=false	Indicates whether or not the ROI metric should include affinity metrics.
promote.partial.threshold=100	Use this parameter to set a forecast ratio threshold for a partial forecast. When the forecast ratio is greater than the configured threshold, the forecast status will reflect <i>Partial, Acceptable</i> . When the forecast ratio is lesser than the configured threshold, the forecast status will reflect <i>Partial, Unacceptable</i> .
promote.forecast.storebase=true	Use this parameter to specify whether the results of Promote Calc Engine's (PCE) store base coverage feature be used in the Promotion Planning and Optimization (PPO) forecast status and forecast ratio. When used, the PCE will provide the store base used in the prediction, so that the application accurately reflects the forecast status, ratio, and coverage.
promote.forecast.getlist=false	Indicates whether or not list SKUs are forecasted (not supported by PCE).
promote.userforecast.empty.offers=false	Sales and margin values can be calculated for offers without any SKUs. This parameter enables the forecasts to be run for empty offers.
promote.cache.expiration=1	The time in minutes before cached data expires. The value of -1 is the default. This value indicates that the cached data never expires.
promote.clear.cache.gc=true	Indicates whether or not the garbage collector is invoked when the cached data is removed.
promote.confidence.greenThreshold=70	Amount (in percentage) of forecast confidence to qualify for a high confidence in the forecast.
promote.confidence.yellowThreshold=30	Amount (in percentage) of forecast confidence to qualify for a medium confidence in the forecast.
promote.confidence.redThreshold=0	Amount (in percentage) of forecast confidence to qualify for a low confidence in the forecast.
promote.show.excluded.items=true	Indicates whether or not excluded items are shown in offer details SKU view.
promote.show.ineffective.items=true	Indicates whether or not ineffective items are shown in offer details SKU view.
promote.offerrule.round=true	Use this parameter to enable the rounding rules for the offer versions.
promote.offerrule.exclude=true	Indicates whether or not new discount price offer exclusion rules are enabled.
promote.offerrule.versions=	Indicates which versions the new discount price offer exclusion rule applies to. This required promote.offerrule.exclude to be true.
promote.created.phase.editable=false	Use this parameter to allow the users to edit the Created phase details.
promote.planning.phase.editable=true	Use this parameter to allow the users to edit the Planning phase details.

Table 4–11 (Cont.) Data Properties

Property	Description
promote.reviewing.phase.editable=false	Use this parameter to allow the users to edit the Reviewing phase details.
promote.revsing.phase.editable=false	Use this parameter to allow the users to edit the Revising phase details.
promote.finalizing.phase.editable=false	Use this parameter to allow the users to edit the Finalizing phase details.
promote.auditing.phase.editable=false	Use this parameter to allow the users to edit the Auditing phase details.
promote.running.phase.editable=false	Use this parameter to allow the users to edit the Running phase details.
promote.completed.phase.editable=false	Use this parameter to allow the users to edit the Completed phase details.
promote.offer.submit.change.invalidation=true	Use this parameter to allow the submit offer status changes to invalidate the offer.
promote.offer.approve.change.invalidation=true	Use this parameter to allow the approve offer status changes to invalidate the offer.
promote.workflow.all.assignments.complete.task=true	Use this parameter to allow workflow tasks to complete for all users when one user completes a task.
promote.workflow.strict.owner=true	Use this parameter to allow the users to reset a phase to in-process.

The value index metric properties define the metric properties.

Table 4–12 Value Index Metric Properties

Property	Description
promote.sales.ratio=0.4d	Sales ratio.
promote.margin.ratio=0.4d	Margin ratio.
promote.units.ratio=0.2d	Units ratio.
promote.affinity.sales.ratio=0.4d	Affinity sales ratio.
promote.affinity.margin.ratio=0.4d	Affinity margin ratio.
promote.affinity.units.ratio=0.2d	Affinity units ratio.
promote.valueidx.offset=100	Value idx offset.

The page properties define page size properties.

Table 4–13 Page Properties

Property	Description
promote.ratio.width=0.20d	Value must be valid Java double between 0 and 1.
promote.ratio.height=0.20d	Value must be valid Java double between 0 and 1.

The position renumbering properties define the position renumbering properties.

Table 4–14 Position Renumbering Properties

Property	Description
promote.page.renumberTopDown	Sets the renumbering to sort based on "y,x" (Top and Down) or "x,y" co-ordinates. The default value is false and indicates that the positions will be renumbered based on x,y co-ordinates.
promote.page.renumberRounding	Sets the decimal point rounding for the X/Y axes co-ordinates. The default value is 1 and indicates that the co-ordinates will be rounded to one digit after the decimal point.

The reports properties define MicroStrategy access.

Table 4–15 Reports Properties

Property	Description
promote.report.auto_auth=false	Separate login for access to MicroStrategy.

The sessions properties define the duration of the keepalive and sessionless timeout.

Table 4–16 Sessions Properties

Property	Description
promote.keepalive.duration=60	Value in minutes for duration of keepalive.
promote.sessionless.timeout=3600	Value in minutes for the sessionless timeout to occur.

The sessions properties define the duration of the keepalive.

Table 4–17 Sessions Properties

Property	Description
promote.keepalive.duration=60	Value in minutes for duration of keepalive.

The following properties define the column/row configuration:

Table 4–18 Column/Row Configuration

Property	Description
columns.Calendar	Columns that appear in the Calendar list.
columns.promotion.versions.PromotionLocationVO	Columns that appear in the Version Groups Detail window for locations.
columns.promotion.versions.PromotionStoreSubSetVO	Columns that appear in the Version Groups Detail window for store subsets.
sort.version.storesubsets	Sort order to be used for the store subsets within the Version Groups Detail window.
columns.Vehicles	Columns that appear in the Vehicles tab of the Promotion Manager.
columns.versionGroups	Columns that appear in the Version Groups user interface.
columns.template.versionGroups	Columns that appear in the Version Groups tab in the Vehicle Template window.

Table 4–18 (Cont.) Column/Row Configuration

Property	Description
columns.promotion.versions.PromotionVehicleLocationVO	Columns that appear in the Version Groups Detail window on the Vehicle Details screen for locations.
columns.promotion.versions.PromotionVehicleStoreSubSetVO	Columns that appear in the Version Groups Detail window on the Vehicle Details screen for store subsets.
columns.VendorDeals	Columns that appear in the Vendor Deals tab of the Promotion Manager.
sort.VendorDeals	Sort order to be used for the entries that appear in the Vendor Deals tab.
columns.Categories	Columns that appear in the Categories tab of the Promotion Manager.
sort.Categories	Sort order to be used for the entries that appear in the Categories tab.

The following properties define the offers configuration:

Table 4–19 Offers Configuration

Property	Description
columns.Offers	Columns that appear in the Offers tab of the Promotion Manager.
sort.Offers	Sort order to be used for the entries that appear in the Offers tab.
columns.copy.Offers	Columns that appear in the offers list window when users try copying an offer from an existing promotion.
sort.copy.Offers	Sort order to be used for the entries that appear in the offers list window.
userDefinedFields.promotion.Offer.Notes.column1	Use these parameter to set up additional user-defined fields/columns in the Notes tab that appears on the Offer Details window.
userDefinedFields.promotion.Offer.Notes.column2	<p>You can add up to eight fields of the integer, decimals, text, boolean, date, or enum types. Fields available for use are integer1, integer2, integer3, integer4, integer5, integer6, integer7, integer8, decimal1, decimal2, decimal3, decimal4, decimal5, decimal6, decimal7, decimal8, text1, text2, text3, text4, text5, text6, text7, text8, boolean1, boolean2, boolean3, boolean4, boolean5, boolean6, boolean7, boolean8, date1, date2, date3, date4, date5, date6, date7, date8, enum1, enum2, enum3, enum4, enum5, enum6, enum7, enum8.</p> <p>In case you specify user-defined columns, you must also specify the relevant field label name in the PromoteResources.properties file.</p> <p>For example to add a 3 new text fields (text1, text2, and text3), specify the configuration as:</p> <ul style="list-style-type: none"> ■ userDefinedFields.promotion.Offer.Notes.column1=text1,text2 ■ userDefinedFields.promotion.Offer.Notes.column2=text3 <p>And to specify the labels "Line Item 1", "Line Item 2", and "Line Item 3", add these lines to the Offer labels section in the PromoteResources.properties file:</p> <ul style="list-style-type: none"> ■ label.promotion.OfferDetailVO.text1=Line Item 1 ■ label.promotion.OfferDetailVO.text2=Line Item 2 ■ label.promotion.OfferDetailVO.text3=Line Item 3
columns.offer.Criteria	Columns that appear in the Criteria View of the Offer Definitions tab.
columns.offer.Criteria.descriptionLevel	The parent hierarchy level of the category or SKU criteria that appears in the Description column of the Criteria View. There are 15 hierarchy levels (valid values are 1 through 15). Setting the value to 1 will display all parents, where as setting the value to 15 will display the category or SKU description.
columns.offer.Items	Columns that appear in the SKU View of the Offer Definitions tab.
columns.offerList.Versions	Columns that appear in the Versions pop-up window on the Offers tab.

Table 4–19 (Cont.) Offers Configuration

Property	Description
columns.offerList. .VersionGroups	Columns that appear in the Version Groups pop-up window on the Offers tab.
columns.offer.Versions	Columns that appear in the Version View of the Offer Definitions tab.
columns.offer.VersionGroups	Columns that appear in the Version Groups View of the Offer Definitions tab.
columns.offerList.Positions	Columns that appear in the Positions pop-up window on the Offers tab.
columns.offer.Positions	Columns that appear in the Position View of the Offer Definitions tab.
columns.offer.criteria.Items	Columns that appear in the Items pop-up window from the Criteria View of the Offer Definitions tab.
columns.Offer.WhatIf	Columns that appear in the List View of the Performance & What-if tab.
sort.Offer.WhatIf	Sort order to be used in the List View of the Performance & What-if tab.
offer.whatif.view	Default view for the Performance & What-if tab. Set the value to 1 for Vertical View and 0 for List View.
promote.offer.setup.scenarios	Default number of scenarios that must appear in the Scenario Setup window.
columns.offer.Affinity	Columns that appear in the Offer Affinity window.
sort.offer.Affinity	Sort order to be used in the Offer Affinity window.
rows.OfferPerformance	Performance metrics rows that appear in the Vertical view on the Performance & What-if tab.
width.OfferPerformance.label	Width of the performance metrics name.
width.OfferPerformance.value	Width of the value for the performance metrics entry.
height.OfferPerformance.metric	Height of the performance metrics row.

The following properties define the vehicle design configuration:

Table 4–20 Vehicle Design Configuration

Property	Description
promotion.vehicleDesign.view	Default view for the Vehicle Design tab. Set the value to 1 for the List View, 2 for the Layout View, or 3 for the Chart View.
columns.vehicleDesign.AllPages	Columns that appear in the All Pages - List View of the Vehicle Design tab.
sort.vehicleDesign.AllPages	Sort order to be used on the All Pages - List View of the Vehicle Design tab.
columns.vehicleDesign.AllPages. .thumbnail	Columns that appear in the All Pages - Thumbnail View of the Vehicle Design tab.
columns.vehicleDesign.Page.position. .thumbnail	Columns that appear in the Single Page - Thumbnail View of the Vehicle Design tab.
columns.vehicleDesign.Page.positions	Columns that appear in the Single Page - List View of the Vehicle Design tab.
columns.vehicleDesign.position.Offers	Columns that appear in the Offers tab of the Position Details window.
columns.vehicleDesign.position.Categories	Columns that appear in the Categories tab of the Position Details window.

The following properties define the Workflow configuration:

Table 4–21 Workflow Configuration

Property	Description
columns.Workflow	Columns that appear in the Workflow tab of the Promotion Manager.
columns.promotion.workflow.Assignments	Columns that appear in the Assignments section that appear in the Task Details window.

The following properties define the Promotion Performance window configuration:

Table 4–22 Promotion Performance Configuration

Property	Description
rows.PromotionPerformance	Rows that appear in the Promotion Performance window.

The following properties define the column/row configuration:

Table 4–23 Category Allocation Configuration

Property	Description
columns.CategoryAllocation	Columns that appear in the Target Allocation window.

The following properties define the find or quick add configuration:

Table 4–24 Finder/Quick Add Configuration

Property	Description
columns.search.Campaigns	Columns that appear in the Campaign Search window.
columns.search.Images	Columns that appear in the Images Search window.
columns.search.Users	Columns that appear in the Users Search window.
columns.search. .SkuListCriteriaQuickAdd	Columns that appear in the search window for the SKU list criteria.
columns.search. .MerchandiseCriteriaQuickAdd	Columns that appear in the search window for the merchandise criteria.
columns.search.LocationsQuickAdd	Columns that appear in the search window for the location criteria.
columns.search.StoreSetsQuickAdd	Columns that appear in the search window for the store sets criteria

The following properties define the merchandise configuration:

Table 4–25 Merchandise Configuration

Property	Description
columns.search.Merchandise	Columns that appear in the search window when users search for a merchandise from the hierarchies selector.
columns.search.SkuLists	Columns that appear in the search window when users search for a SKU from the hierarchies selector.
columns.search.Location	Columns that appear in the search window when users search for a location from the hierarchies selector.
tree.storeset.sort	Sort order to be used in the expandible/collapsible tree in the hierarchies selector for store sets.

Table 4–25 (Cont.) Merchandise Configuration

Property	Description
tree.storeset.label.parts	Format of the store set name that appears in the hierarchies selector. The name format defaults to the store set ID followed by the store set name.
columns.search.storeset	Columns that appear in the search window when users search for a store set from the hierarchies selector.
columns.search.storesubset	Columns that appear in the search window when users search for a store subset from the hierarchies selector.
columns.setup.SkuListItems	Columns that appear in the SKU list pop-up window.
tree.label.parts	Format of the merchandise name that appears in the hierarchies selector. The name format defaults to the merchandise ID followed by the merchandise name.
tree.label	Label for the expandible/collapsible tree that displays the merchandise hierarchy.
tree.sort	Sort used in the expandible/collapsible tree.
tree.lookAheadPages	Number of look ahead pages set for the expandible/collapsible tree.
tree.loadingLabel	The loading label used for the expandible/collapsible tree.
tree.indentation	Indentation used for the expandible/collapsible tree.

The following properties define the tasks configuration:

Table 4–26 ToDo Configuration

Property	Description
columns.todo.MyTasks	Columns that appear in the My Tasks window.
columns.todo.MyApprovals	Columns that appear in the My Approvals window.

The following properties define the templates setup configuration:

Table 4–27 Setup Configuration

Property	Description
columns.setup.PageTemplates	Columns that appear in the Page Templates window.
columns.setup.PageTemplatePositions	Columns that appear in the List View of the Positions tab in the Page Templates window.
columns.setup.VehicleTemplates	Columns that appear in the Vehicle Templates window.
columns.template.versions .VehicleTemplateLocationVO	Columns that appear for locations in the Version Groups tab of the Vehicle Templates window.
columns.template.versions .VehicleTemplateStoreSubSetVO	Columns that appear for the store subsets in the Version Groups tab of the Vehicle Templates window.
columns.setup.VehicleTemplatePages	Columns that appear in the List View of the Vehicle Design tab on the Vehicle Templates window.
columns.setup.PromotionTemplates	Columns that appear in the Promotion Templates window.
columns.template.versions .PromotionTemplateLocationVO	Columns that appear for locations in the Version Groups tab of the Promotion Templates window.
columns.template.versions .PromotionTemplateStoreSubSetVO	Columns that appear for the store subsets in the Version Groups tab of the Promotion Templates window.

Table 4–27 (Cont.) Setup Configuration

Property	Description
columns.setup .PromotionTemplateVehicles	Columns that appear in the Vehicles tab of the Promotion Templates window.
columns.setup .PromotionTemplateWorkflow	Columns that appear in the Workflow tab of the Promotion Templates window.
columns.setup .PromotionTemplateWorkflowAssignments	Columns that appear in the Assignments section that appear in the Task Details window.

The following properties define the list configuration:

Table 4–28 List Configuration

Property	Description
list.buttons.copy	Use this parameter (set the value to true) to enable the Copy feature in the List View of the Performance & What-if tab.
list.buttons.export	Use this parameter (set the value to true) to enable the Export feature in the List View of the Performance & What-if tab.
list.buttons.print	Use this parameter (set the value to true) to enable the Print feature in the List View of the Performance & What-if tab.
list.header.maxHeight	Maximum height for a list header.

The following properties define the chart configuration:

Table 4–29 Chart Configuration Parameters

Property	Description
performanceChart.promotion .columnChartOptions	Types of column charts available for promotions. By default, the % Performance (percentColumnChart) and Amount/Units (amountUnitsColumnChart) column charts are configured and available.
performanceChart.promotion .bubbleChartOptions	Types of bubble charts that available for promotions. By default, the % Performance (percentBubbleChart) bubble chart is configured and available.
performanceChart.offer .columnChartOptions	Types of column charts available for offers. By default, the % Performance (percentColumnChart) and Amount/Units (amountUnitsColumnChart) column charts are configured and available.
performanceChart.offer .bubbleChartOptions	Types of bubble charts that available for offers. By default, the % Performance (percentBubbleChart) bubble chart is configured and available.
performanceChart.page .columnChartOptions	Types of column charts available for pages. By default, the % Performance (percentColumnChart) and Amount/Units (amountUnitsColumnChart) column charts are configured and available.
performanceChart.page .bubbleChartOptions	Types of bubble charts that available for pages. By default, the % Performance (percentBubbleChart) bubble chart is configured and available.
performanceChart.categories .columnChartOptions	Types of column charts available for categories. By default, the % Performance (percentColumnChart) and Amount/Units (amountUnitsColumnChart) column charts are configured and available.

Table 4–29 (Cont.) Chart Configuration Parameters

Property	Description
performanceChart.categories.bubbleChartOptions	Types of bubble charts that available for categories. By default, the % Performance (percentBubbleChart) bubble chart is configured and available.
performanceChart.defaultRangeMin	The default minimum range for the charts.
performanceChart.defaultRangeMax	The default maximum range for the charts.
performanceChart.pageSize	Page size for the charts.
performanceChart.maxLabelLength	The maximum length of the chart labels.
performanceChart.percentColumnChart.numColumnSets	Number of column sets used in the % Performance column chart. You must set up the metrics, colors, and Y axes values for each column set. A set of parameters for 3 column sets are available by default. In case you add or modify the number of column sets, ensure that the relevant set of parameters required for each column set is also updated.
performanceChart.percentColumnChart.columnSet1.metrics	Name of the metric used in the column set 1. For the % Performance chart, the value is marginLift.
performanceChart.percentColumnChart.columnSet1.colors	Color for the metric used in the column set 1.
performanceChart.percentColumnChart.columnSet1.yAxis	The Y axis used for column set 1. In a Column Chart, two Y axes are available so that columns with different scales can be displayed. Valid values are 1 or 2 for each column set.
performanceChart.percentColumnChart.columnSet2.metrics	Name of the metric used in the column set 2. For the % Performance chart, the value is salesLift.
performanceChart.percentColumnChart.columnSet2.colors	Color for the metric used in the column set 2.
performanceChart.percentColumnChart.columnSet2.yAxis	The Y axis used for column set 2. In a Column Chart, two Y axes are available so that columns with different scales can be displayed. Valid values are 1 or 2 for each column set.
performanceChart.percentColumnChart.columnSet3.metrics	Name of the metric used in the column set 3. For the % Performance chart, the value is lift.
performanceChart.percentColumnChart.columnSet3.colors	Color for the metric used in the column set 3.
performanceChart.percentColumnChart.columnSet3.yAxis	The Y axis used for column set 3. In a Column Chart, two Y axes are available so that columns with different scales can be displayed. Valid values are 1 or 2 for each column set.
performanceChart.percentColumnChart.yAxis1.format	Format of the first Y axis.
performanceChart.percentColumnChart.yAxis2.format	Format of the second Y axis.
performanceChart.percentColumnChart.filters	Appearance of the Column Chart can be changed by filtering the information in the metrics specified in this parameter.

Table 4–29 (Cont.) Chart Configuration Parameters

Property	Description
performanceChart.amountUnitsColumnChart.numColumnSets	Number of column sets used in the % Performance column chart. You must set up the metrics, colors, and Y axes values for each column set. A set of parameters for 3 column sets are available by default. In case you add or modify the number of column sets, ensure that the relevant set of parameters required for each column set is also updated.
performanceChart.amountUnitsColumnChart.columnSet1.metrics	Names of the metrics used in the column set 1. For the Amount/Units chart, the values is incrementalMargin,affinityIncrementalMargin.
performanceChart.amountUnitsColumnChart.columnSet1.colors	Color for the metrics used in the column set 1.
performanceChart.amountUnitsColumnChart.columnSet1.yAxis	The Y axis used for column set 1. In a Column Chart, two Y axes are available so that columns with different scales can be displayed. Valid values are 1 or 2 for each column set.
performanceChart.amountUnitsColumnChart.columnSet2.metrics	Names of the metrics used in the column set 2. For the Amount/Units chart, the value is incrementalSales,affinityIncrementalSales.
performanceChart.amountUnitsColumnChart.columnSet2.colors	Colors for the metrics used in the column set 2.
performanceChart.amountUnitsColumnChart.columnSet2.yAxis	The Y axis used for column set 2. In a Column Chart, two Y axes are available so that columns with different scales can be displayed. Valid values are 1 or 2 for each column set.
performanceChart.amountUnitsColumnChart.columnSet3.metrics	Names of the metrics used in the column set 3. For the Amount/Units chart, the value is incrementalUnits,affinityIncrementalUnits.
performanceChart.amountUnitsColumnChart.columnSet3.colors	Colors for the metrics used in the column set 3.
performanceChart.amountUnitsColumnChart.columnSet3.yAxis	The Y axis used for column set 3. In a Column Chart, two Y axes are available so that columns with different scales can be displayed. Valid values are 1 or 2 for each column set.
performanceChart.amountUnitsColumnChart.yAxis1.format	Format of the first Y axis.
performanceChart.amountUnitsColumnChart.yAxis2.format	Format of the second Y axis.
performanceChart.amountUnitsColumnChart.filters	Appearance of the Column Chart can be changed by filtering the information in the metrics specified in this parameter.
performanceChart.percentBubbleChart.xAxis.metric	Name of the metric used in the X axis of the Bubble Chart. For the % Performance bubble chart, the value is salesLift.
performanceChart.percentBubbleChart.xAxis.color	Color for the metric used in the X axis of the Bubble Chart.
performanceChart.percentBubbleChart.xAxis.format	Format of the X axis of the Bubble Chart.
performanceChart.percentBubbleChart.yAxis.metric	Name of the metric used in the Y axis of the Bubble chart. For the % Performance bubble chart, the value is marginLift.

Table 4–29 (Cont.) Chart Configuration Parameters

Property	Description
performanceChart.percentBubbleChart.yAxis.color	Color for the metric used in the Y axis of the Bubble Chart.
performanceChart.percentBubbleChart.yAxis.format	Format of the Y axis of the Bubble Chart.
performanceChart.percentBubbleChart.bubble.metric	Name of the metric used in the bubble of the Bubble Chart. For the % Performance bubble chart, the value is totalSales.
performanceChart.percentBubbleChart.bubble.color	Color for the metric used in the bubble of the Bubble Chart.
performanceChart.percentBubbleChart.filters	Appearance of the Bubble Chart can be changed by filtering the information in the metrics specified in this parameter.

PPO Configuration File

GUI-wide properties are set in the promote.properties file, which is located in <configroot>/promote. This file is pre-populated during the installation process with installation-specific values. The complete list of properties in promote.properties can be found in the Configuration Guide.

Other properties relate to OAS 10.1.3.3 and contain the prefix oas. For example, oas.java.naming.provider.url has to be set to the correct (opmn or non-opmn) url, such as ormi://host:port.

The same is true for the configuration of the User Management application. Its properties are stored in <configroot>/usermanagement/usermanagement.properties.

To specify the time interval at which PI-PPO Planning updates the server session on browser-only user activity (that is, when a user click does not result in a server call), set promotekeepalive.interval to the desired interval (in seconds).

Configuring Display Strings

GUI resources such as labels and error messages are kept in the promoteResources.properties file, which is located in <configroot>/promote.

Note that all internationalization configuration settings are contained in promote.properties and all translated text for display in the UI are contained in promoteResources.properties.

The promoteResources.properties file is organized into functional sections, most of which define information presented to the user that should not be modified. Each section is preceded by a comment that defines either the purpose of the section (such as Error Messages) or the screen in the UI that the section details (such as Promotion Template).

The promoteResources.properties file also provides limited functionality to configure the columns and rows that appear in certain screens. Consult this file for more information about configuring columns and rows.

You can configure the following:

- Which columns or rows are displayed.
- The sort order, descending (-) or ascending (+), of specified columns. For example, sort.Offers=+position sorts the grid in ascending order based on the position column.

- Which metrics are displayed. You can select from two lists of available metrics: `BASE_METRIC_COLUMNS`, which is a list of common metrics, and `METRIC_COLUMNS`, which is a list of additional available metrics.
- Column locking (defined using a pipe symbol).
- User defined fields can be added to the Offer Definition and Notes grids. Different types of fields, such as text or date) can be selected and client-defined labels can be specified for the fields.

Configuring Export

The following style sheets are shipped with PI-PPO Planning:

- `XmlExportTemplate.xslt`, which is used to format the XML output of a promotion
- `TxtExportTemplate.xslt`, which describes the instructions for the TXT format.
- `PromoOfferItemSumTxtTmpl.xslt`, which provides promotion offer item details.
- `PromoOfferPosSumTxtTmpl.xslt`, which provides offer position details.

The location and naming of these files are specified in the `promote.properties` file, which is located in `<configroot>/promote`.

The following values must be specified:

Table 4–30 Export Configuration Values

Value	Description
export.root.path	Location of exported files for a push export
export.xml.template	Location of the XML format XSLT style sheet (e.g., <configroot>/config/promote/XmlExportTemplate.xslt)
export.txt.template	Location of the TXT format XSLT style sheet (e.g., <configroot>/config/promote/TxtExportTemplate.xslt)

Values for export.txt.template must be specified for all export types:

- promote.export.txt.template.promotion
- promote.export.txt.template.promotionofferitemssummary
- promote.export.txt.template.promotionofferpositionssummary

Pull Export Configuration

Two files must be configured for a pull export:

- promo-pullclient.properties – defines the defaults for the pull client
- promo-pullclient.log4j.properties – defines the Log4j configuration

These files are located in <installdir>/modules/tools/conf.

Example promo-pullclient.properties File

```
promote.pullclient.servlet.contextroot=promote
promote.pullclient.servlet.appname=export.do
promote.pullclient.protocol=http
promote.pullclient.host=localhost
promote.pullclient.port=8888
promote.pullclient.datemask=MM/dd/yyyy_HH:mm:ss
promote.pullclient.format=xml
promote.pullclient.command=list
promote.pullclient.timeout=10
```

No spaces are permitted for any of the assigned values. The date mask specifies only the input arguments format. The output format is specified in promote.properties.

Integration with Promotion Intelligence

The following configuration points must be set so that Promotion Intelligence reports can be open from PI-PPO Planning.

Auto Authentication Flag

The `promote.properties` file contains an auto-authentication flag called `promote.report.auto_auth`. The values for the flag are **true** and **false**.

When the flag is set to true, the Promo Planning/Intelligence integration uses the currently logged-in user's name and password when logging into MicroStrategy.

Report Links Configuration

Report mapping and report links must be defined in `<configroot>/promote/promote-config.xml`. A sample file is populated during the installation procedure. The XML schema definition file is located in `<OAS-dir>/j2ee/home/applications/promote/xmlSchema/promote.xsd`.

This configuration includes:

- The MicroStrategy server DNS name, port, protocol, and webapp name
- The organization of the MicroStrategy reports into groups and the list of reports that are included in each group
- Label displays
- Resource file mapping information

The following XML attributes are used in `promote.xml`:

- Connect attributes used in the construction of the URL for all links (protocol://server:port/webapp_path).
- Each reporting group has its own node. The name is used for the resource file mapping. The report request uses the `param` tag.
- For all report tags in group, sub-nodes are created in the GUI for the reporting area, using name, params and the common configuration from connect tag. A group with no reports does not have sub-nodes.
- If there is no `params` tag inside the group or report tag, then no link is provided.
- Groups cannot be nested inside other groups or reports.

Display Strings

The Promotion Planning and Promotion Intelligence GUI properties are located in `promoteResources.properties`. This file includes locale-specific labels and descriptions. The value name in `promote-config.xml` is used as the key in the resource file.

Here is an example, using "My Reports":

```
label.report.MyReports.name=My Reports
```

```
label.report.SharedReports.name=Shared Reports
```

```
label.report.SharedReports.AdPageAllocation.name=Ad Page Allocation
```

```
label.report.SharedReports.AdRoi.name=Ad ROI
```

Debug Messages

The log files are located in `<configroot>/promote/promote.log4j.properties`. The location of the file and the debug level can both be modified. If changes are made to these values, the application server must be restarted.

Staging and Loading Data

This chapter provides details about the staging and loading of data into the PI-PPO application. For details about the standard interface as well as configuration details, see the *Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization Operations Guide*.

This chapter contains the following sections:

- “Data Standard Interface” on page 5-1
- “Standard Load Process” on page 5-1
- “Standard Load Error Handling” on page 5-3
- “Standard Load Procedures Order” on page 5-4

Data Standard Interface

The standard interface chapter of the *Oracle Retail Promotion Intelligence and Promotion Planning and Optimization Operations Guide* discusses the specific data and formatting requirements for the standard interface. Use these interface descriptions as the model for creating the required text files for the database.

Standard Load Process

The application provides two scripts that manage the process to stage, transform, and load data into the target database tables in the database. The data must be provided in flat files that meet the standard interface specifications. The variable length data in the files should be pipe-delimited. The files should be named to correspond to the names of the matching specification tables. For example, the calendar file should be named in a meaningful way (such as cal.txt) to correspond to ASH_CAL_TBL. No specific file extension is required for the input files.

The two scripts are located in %INSTALLATION_DIRECTORY%/modules/tools/bin. The first script, **pl_stage_file.sh**, stages the data from the flat files into the staging tables. The second script, **pl_load_data.sh**, loads the staged data into the database. These two scripts are used if you need to customize the load dependency tree.

Each script contains options that can be customized. You can customize the options in the following ways (which are listed in order of precedence, with the command line having the highest precedence):

- Using the command line options
- Setting the customization values as environment variables in env.sh
- Setting the customization values in the user’s environment

If you do not need to customize the load dependency tree, you can use the following two scripts:

- **pl_stage_client.sh** *<full_path_to_product_directory> DatasetFilename*
- **pl_load_client.sh** *<full_path_to_product_directory>*

The **pl_stage_client.sh** script calls **pl_stage_file.sh**. The **pl_load_client.sh** script calls **pl_load_data.sh**.

Environment Customization File

Here is an example of the environment customization file (**env.sh**):

```
#This is the environment customization file.
#Please define all customization values here.

#The mail client and address to send all messages to:
#MAIL=mailx
#REPORT_ADDRESS=error_mail@your_domain.com

#Number of parallel processes to run load procedures:
PARALLEL=2

#Directory with data control files:
#CONTROLDIR=/ASHschema/controlfiles

#Directory to store logs:
#LOGDIR=/tmp/load_logs

#Directory to move old logs to.
#If this variable is not set, the logs will be overwritten.
This folder is not required to exist and will be created at the time
#of archiving the logs.
#
#If all old logs should be preserved, it is possible to
#archive the files into a new unique folder, such as:
#LOGDIR_ARCHIVE=
#/tmp/load_logs/archived_logs_'date +%Y%m%d_%H%M%S'
#
#If only the archive of the previous run is important, then
#archive the files into the same folder, such as:
#LOGDIR=/tmp/load_logs/archived_logs

#Number of errors to allow during load
ERROR_THRESHOLD=50
```

Staging Script: pl_stage_file.sh

Usage: **pl_stage_file.sh** [OPTION]... [FILE]...

Loads the files into the database.

Options:

Table 5–1 *pl_stage_file.sh* Options

-a DIR	--logdir_archive=DIR	directory to archive old log files
-c DIR	--controldir=DIR	directory with data control files

Table 5–1 (Cont.) pl_stage_file.sh Options

-e NUM	--errorthreshold=NUM	number of errors to allow in load (for DB2, it is a warning threshold)
-l DIR	--logdir=DIR	directory to store logs
-r DIR	--configroot=DIR	configuration root directory
-f FILENAME	--paramfile=FILENAME	filename of DB parameter file in configroot
-h	--help	displays help and exits

Load Script: pl_load_data.sh

Usage: `pl_load_data.sh [OPTION]... [LOADPROCEDURE]...`

Runs the load procedures in the database.

Options:

Table 5–2 pl_load_data.sh Options

-a DIR	--logdir_archive=DIR	directory to archive old log files
-e NUM	--errorthreshold=NUM	number of errors to allow in load (overwrites the procedure's default limit)
-l DIR	--logdir=DIR	directory to store logs
-r DIR	--configroot=DIR	configuration root directory
-f FILENAME	--paramfile=FILENAME	filename of DB parameter file in configroot
-rlo	--runloadonly	do not run pre and post step
-ne	--noexit	does not exit with a value
-h	--help	displays help and exits

Standard Load Error Handling

The Standard Load verifies the records in each staging table. Each record that fails the verification is removed from the staging table and placed in another table so that the load can continue and so that the failed records can be reviewed.

If a load procedure fails and the threshold is exceeded, you will see the message “The specified error threshold has been exceeded for this load procedure.” If this occurs, you should correct the existing data problem and re-run the load procedure as well as any child load procedures (as shown in [“Standard Load Procedures Order” on page 5-4](#)).

The table containing the failed records is assigned a name that corresponds to the associated staging table. For example:

Table 5–3 Failed Records Table Names

Staging Table	Failed Record Table
ASH_CAL_TBL	ASH_CAL_TBL_BAD
BEECH_OFFER	BEECH_OFFER_BAD

The “BAD” table into which the failed records are inserted has the same structure as the corresponding staging table with the addition of the following four columns:

Table 5–4 Bad Table Columns

Column Name	Description	Data Type	Maximum Length	Nullable (Y/N)
ERROR_ROWID	The row ID that corresponds to the row ID in the staging table	Row ID		N
ERROR_CODE	The code for the verification	Integer		N
ERROR_DESC	Description of the error	String	1000	
ERROR_TIME	The time the error occurred	Timestamp		N

It is possible to place a threshold on the number of failed records in any staging table that will trigger a termination of the load. The default threshold values are hard-coded into the application. In order to customize the threshold values, you must create a properties file and load it into the application.

Standard Load Procedures Order

The standard load should execute in the following order:

1. The staging of:
 - ASH_CAL_TBL
 - BEE_PERIODS_ATTR_TBL
 - ASH_CP_TBL
 - ASH_LHL_TBL
 - ASH_MHL_TBL
 - ASH_LHRENAME_TBL
 - ASH_LH_TBL
 - ASH_MHRENAME_TBL
 - ASH_MH_TBL
 - BEE_IMAGE
 - BEE_OFFER
 - BEE_USER_DEFINED_TYPE
 - BEE_USER_DEFINED_VALUE
 - BEE_VEHICLE_ATTR
 - BEE_VEHICLE
 - CLIENT_HIERARCHY_ACTIONS_TBL
 - STAGE_MH_ATTRS_TBL
 - ASH_LH_ATTRS_TBL
 - BEE_STORE_SETS
 - BEE_STORE_SUBSETS
 - BEE_STORE_SUBSET_ASGN

- BEE_STORE_SET_PRICE
 - BEE_PR_LIKE_LOCATION
 - BEE_PR_LIKE_MERCHANDISE
 - BEE_SKU_LIST
 - BEE_SKU_LIST_ITEM
 - BEE_PROMO_ALLOC
 - BEE_PROMO_CAMPAIGN
 - BEE_PROMO_OFFER_ATTR
 - BEE_PROMO_OFFER_MERCH
 - BEE_PROMO_OFFER
 - BEE_PROMO_STORE
 - BEE_PROMOTIONS
 - BEE_PROMO_FRCSTR
 - ASH_PARAMETER_VALUES_TBL
 - ASH_SEASONALITY_MAPS_TBL
 - ASH_SEASONALITY_VALUES_TBL
 - BEE_FUTURE_PRICE_COST
 - BEE_MB_DETAIL
 - WK_HIST_SALES_INV
2. These load procedures:
- com.profitlogic.db.birch.LoadCalendars
 - com.profitlogic.db.birch.LoadLHKeyRename
 - com.profitlogic.db.birch.LoadMHKeyRename
 - com.profitlogic.db.birch.LoadMerchandiseHierarchy
 - com.profitlogic.db.birch.LoadLocationHierarchy
 - com.profitlogic.db.birch.LoadMHTbl
 - com.profitlogic.db.birch.LoadLHTbl
 - com.profitlogic.db.birch.LoadTClose
 - com.profitlogic.db.birch.LoadLTClose
 - com.profitlogic.db.beech.LoadTypeMaster
 - com.profitlogic.db.beech.LoadValueMaster
 - com.profitlogic.db.beech.LoadOfferMaster
 - com.profitlogic.db.beech.LoadImageMaster
 - com.profitlogic.db.beech.LoadVehicleMaster
 - com.profitlogic.db.beech.LoadVehicleAttributeMaster
 - com.profitlogic.db.birch.LoadMerchandiseAttributes
 - com.profitlogic.db.birch.LoadLHAttributes

- com.profitlogic.db.beech.LoadPromoLikeLocation
- com.profitlogic.db.beech.LoadPromoLikeMerchandise
- com.profitlogic.db.beech.LoadSkuListMaster
- com.profitlogic.db.beech.LoadSkuListItemMaster
- com.profitlogic.db.beech.LoadSkuListCleanupMaster
- com.profitlogic.db.beech.LoadStoreSets
- com.profitlogic.db.beech.LoadStoreSubsets
- com.profitlogic.db.beech.LoadStoreSubsetAssignments
- com.profitlogic.db.beech.LoadStoreSetPrice
- com.profitlogic.db.beech.LoadPromoCampaign
- com.profitlogic.db.beech.LoadHistoricPromo
- com.profitlogic.db.beech.LoadPromoVehicle
- com.profitlogic.db.beech.LoadPromoVehiclePage
- com.profitlogic.db.beech.LoadPromoVehicleLocation
- com.profitlogic.db.beech.LoadPromoVehicleAlloc
- com.profitlogic.db.beech.LoadPromoOffer
- com.profitlogic.db.beech.LoadPromoOfferMerchandise
- com.profitlogic.db.beech.LoadPromoVehiclePagePosOff
- com.profitlogic.db.beech.LoadPromoVehiclePagePositionOfferAttribute
- com.profitlogic.db.beech.LoadPromoForecaster
- com.profitlogic.db.birch.LoadParameters
- com.profitlogic.db.birch.LoadSeasonalities
- com.profitlogic.db.beech.LoadFuturePriceCost
- com.profitlogic.db.walnut.CreateMerchSummaryTbl
- com.profitlogic.db.walnut.CreatemerchPromoMBSummaryTbl
- com.profitlogic.db.walnut.CreateMerchMapView
- load_weekly_history_data.load_at_all_levels
- pr_load_sales_price
- com.profitlogic.db.beech.LoadMbDetail
- pr_promoitem_load
- pr_promolocation_load
- pr_process_mb_detail
- pr_load_promo_mb_summary.load_all
- pr_process_apc_summary
- pr_create_store_count_sum
- pr_analytics_pkg.refresh_effective_price

Standard Load Steps

Each procedure consists of the following sub-procedures:

1. Setup
2. Pre-load Verification. All n processes are run in parallel.
3. Finish Pre-load Verification.
4. Load. All n processes are run in parallel.
5. Post-load Verification. All n processes are run in parallel.
6. Finish Post-load Verification.
7. Tear-down.

Database Configuration

This chapter provides technical details about the database configuration. It contains the following sections:

- “Summary Configurations” on page 6-1
- “CLIENT_HIERARCHY_ACTIONS_TBL” on page 6-4
- “IR Views” on page 6-4
- “PR_DB_PARAMS” on page 6-5

Summary Configurations

Several configurations must be included in ASH_CP_TBL. These configurations specify the level of aggregation in the merchandise hierarchy that the application and the RDM require.

Table 6–1 Summary Configurations

INTERSECT_NAME	MERCHANDISE_LEVEL	LOCATION_LEVEL	Description
PROMOTE_TAE	SKU	DISTRICT	Identifies the Level at which TAE output is produced.
PROMOTE_DETAIL	SKU	STORE	Identifies the Level at which POS data is expected. It is assumed to be the STORE level.
PROMOTE_SUMMARY_1	CLASS	STORE	Identifies the Merchandise and Location levels of the first level of the summary.
PROMOTE_SUMMARY_2	DEPARTMENT	STORE	Identifies the Merchandise and Location levels of the second level of the summary.
PROMOTE_SUMMARY_3	DIVISION	STORE	Identifies the Merchandise and Location levels of the third level of the summary.
PROMOTE_AFFINITY_LEVEL	CLASS	CHAIN	The level of calculation of the APE summary.
PROMOTE_APC	CLASS	REGION	The level of calculation of the APC summary.
PROMOTE_ANALYSIS	SKU	COUNTRY	

Table 6–1 (Cont.) Summary Configurations

INTERSECT_NAME	MERCHANDISE_LEVEL	LOCATION_LEVEL	Description
PROMOTE_SCORECARD_SUMMARY_1	SUBCLASS	STORE	Specifies the level of aggregation from the MH that is used to generate the totals for the scorecard by the MH.
PROMOTE_SCORECARD_SUMMARY_2	CLASS	STORE	Specifies the level of aggregation from the MH that is used to generate the totals for the scorecard by the MH.
PROMOTE_MIN_LCD	DEPT	CHAIN	Defines the lowest level of the hierarchy that is available for display in the UI.
PROMOTE_PROMO_OFFER_MH_SUMMARY	DEPT	STORE	Specifies the level of aggregation from the MH that is used to generate the totals for the Scorecard by Offer/Dept report.
PROMOTE_SCORECARD_MERCH_OFF_AMT_SUMM_3	DEPT	STORE	Specifies the level of aggregation from the MH that is used to generate the totals for the Scorecard by MH and Offer Amt.
PROMOTE_SCORECARD_MERCH_OFF_AMT_SUMM_2	CLASS	STORE	Specifies the level of aggregation from the MH that is used to generate the totals for the Scorecard by MH and Offer Amt.
PROMOTE_SCORECARD_MERCH_OFF_AMT_SUMM_1	SUBCLASS	STORE	Specifies the level of aggregation from the MH that is used to generate the totals for the Scorecard by MH and Offer Amt.
PROMOTE_TAE_NONAD_PART_LEVEL_1	SUBCLASS	CHAIN	Specifies the level of aggregation from the MH that is used for the TAE non-ad metrics. It is also used by the TAE process to identify the starting MH level that should be used to generate its output.
PROMOTE_TAE_NONAD_PART_LEVEL_2	CLASS	CHAIN	Specifies the level of aggregation from the MH that is used for the TAE non-ad metrics. It is also used by the TAE process to identify the starting MH level that should be used to generate its output.
PROMOTE_TAE_NONAD_PART_LEVEL_3	DEPT	CHAIN	Specifies the level of aggregation from the MH that is used for the TAE non-ad metrics. It is also used by the TAE process to identify the starting MH level that should be used to generate its output.
PROMOTE_MIN_BL_AGGR_LEVEL	SUBCLASS	CHAIN	Specifies the lowest level that aggregated baseline data should be calculated for.
PROMOTE_MAX_BL_AGGR_LEVEL	CHAIN	CHAIN	Specifies the highest level that aggregated baseline data should be calculated for.
AE_ANALYSIS	SKU	STORE	

Table 6–1 (Cont.) Summary Configurations

INTERSECT_NAME	MERCHANDISE_LEVEL	LOCATION_LEVEL	Description
PROMOTE_MIN_NONAD_METRIC_AGGR_LEVEL	CLASS	CHAIN	Specifies the lowest level that aggregated Non-Ad metrics should be calculated for.
PROMOTE_MAX_NONAD_METRIC_AGGR_LEVEL	CHAIN	CHAIN	Specifies the highest level that aggregated Non-Ad metrics should be calculated for.
PROMOTE_INVENTORY	SKU	STORE	

The following non-PI and PPO entries are required for compatibility reasons:

Table 6–2 Intersect Names

INTERSECT_NAME	MERCHANDISE_LEVEL	LOCATION_LEVEL
OPTIMIZATION	SKU	STORE
WORKSHEET	DEPARTMENT	CHAIN
SALES	SKU	CHAIN
CLUSTER	CHAIN	CHAIN
DEFAULTLEVEL	CHAIN	CHAIN

The Cust_Parameter_Levels PL/SQL package provides an interface to the following values. For examples, see “IR Views” on page 6-4.

- getMerchandiseLevelDesc(in_intersect_name)
- getMerchandiseLevelSqc(in_intersect_name)
- getLocationLevelDesc(in_intersect_name)
- getLocationLevelSqc(in_intersect_name)

CLIENT_HIERARCHY_ACTIONS_TBL

The Client_Hierarchy_Actions_Tbl must be modified according to the levels of inventory aggregation required.

Table 6–3 Actions for Hierarchy Actions Table

Action Type	Action Name	Action Level Name	Action Level	Hierarchy Type	Description
SUITE	STORE	STORE	0	LOCATION	Identifies the level in the location hierarchy corresponding to physical STORE
PROMOTE	HIST_AGG_MERCH_LEVEL_0	HIST_AGG_LEVEL_0	0	MERCHANDISE	Identifies the Lowest Merchandise Level at which History should be persisted
PROMOTE	HIST_AGG_LOC_LEVEL_0	HIST_AGG_LEVEL_0	0	LOCATION	Identifies the Lowest Location Level at which History should be persisted

IR Views

The following views must be modified according to the level of summary needed. The view creation scripts are located in <installdir>/modules/Database/ROSEWOODSchema/install/oracle/ROSEWOODSchema/dictionary/views_ir. Example (found in the supplied sample KSIInc dataset) are located in <installdir>/modules/pce/sample/ir_views/oracle.

Table 6–4 Modifying Inference Rules

View	Description
IR_OLF_CANDIDATES_VW	This view defines what merchandise nodes are the source of Offer level Forecast (OLF) aggregates.
IR_OLF_NODES_VW	This view defines what merchandise receives an OLF forecast.
IR_PBL_ATTRS_PURCHASE_X_VW	This view defines how the merchandise hierarchy is divided for the predicted baseline calculation.
IR_PR_DEFAULT_PRICE_ZONE_VW	This required view must be manipulated so that it references the primary store set for which pricing data is provide via the Store Set Prices interface. it is used to generate price data for other store sets or location hierarchy levels.
IR_PR_LOCATION_SUMMARY_X_VW	These views map each location summary level to its SKU.
IR_PR_MERCH_SUMMARY_X_VW	These views map each merchandise summary level to it SKU.
IR_PR_PROMO_ITEM_VW	This view exposes the attributes needed by the PCE for modeling.
IR_PR_PROMOTIONS_VW	This view exposes the attributes needed by the PCE for modeling.

Table 6–4 (Cont.) Modifying Inference Rules

View	Description
IR_TREND_CANDIDATES	

Update the views using the following guidelines:

For ir_pr_merch_summary_X_vw. These views map each merchandise summary level to its SKUs. For example:

- CREATE OR REPLACE VIEW ir_pr_merch_summary_3_vw AS SELECT hierarchy3_pid parent_pid, merchandise_id, mod(merchandise_id,10) seas_cd FROM merchandise_tbl WHERE level_sqc = 6
- CREATE OR REPLACE VIEW ir_pr_merch_summary_4_vw AS SELECT hierarchy4_pid parent_pid, merchandise_id, mod(merchandise_id,10) seas_cd FROM merchandise_tbl WHERE level_sqc = 6
- CREATE OR REPLACE VIEW ir_pr_merch_summary_5_vw AS SELECT hierarchy5_pid parent_pid, merchandise_id, mod(merchandise_id,10) seas_cd FROM merchandise_tbl WHERE level_sqc = 6

For ir_pr_location_summary_X_vw. These views map each location summary level to its SKUs. For example:

- CREATE OR REPLACE VIEW ir_pr_location_summary_1_vw AS SELECT hierarchy1_lid, location_id FROM location_tbl WHERE level_sqc = Cust_Parameter_Levels.getLocationLevelsqc('PROMOTE_ANALYSIS')
- CREATE OR REPLACE VIEW ir_pr_location_summary_7_vw AS SELECT hierarchy7_lid parent_lid, location_id FROM location_tbl WHERE level_sqc = Cust_Parameter_Levels.getLocationLevelsqc('PROMOTE_ANALYSIS')

PR_DB_PARAMS

When PPO and RDF are deployed in the same environment, each application uses the base demand forecast generated by RDF for forecasting. The field PREDICT_BASELINE_SOURCE_TYPE, which must be removed if the value generated by RDF is not used is part of the PR_DB_PARAMS table shown in [Table 6–5, "PR_DB_PARAMS"](#).

Table 6–5 PR_DB_PARAMS

Field Name	Description
LAST_CREATE_MISSING_PROMO_CT	The date of the last attempt to create missing promotion counts. Only promotions modified after this date can be fixed when the promotion counts are created again.
LAST_PROCESSED_MB_DATE	The last market basket date loaded.
MB_START_DATE	Obsolete.
MB_END_DATE	Obsolete.
DEFAULT_TABLESPACE	The tablespace that holds the tables created by the application.
DEFAULT_INDEX_TABLESPACE	The tablespace that holds the indexes created by the application.
PROMOTE_SCORECARD_TOP_NONOD	The number of records stored by the TAE NonAd contributor feature. This affects Scorecard reports.

Table 6–5 (Cont.) PR_DB_PARAMS

Field Name	Description
APE_DFLT_NODE_DESCR	The name of the default APE node that receives miscellaneous affinity numbers during forecasting.
LAST_LOAD_MISSING_PROMO_SUM	The date of the last attempt to create missing promotion summaries. Only promotions modified after this date can be fixed when the promotion summaries are created again.
PREDICT_BASELINE_SOURCE_TYPE	The source data for externally provided predicted baseline values. This field must be removed if the external predict baseline is not supported.

Managing the Environment

This chapter provides reference material for systems administrators and others responsible for maintaining the Promotion Intelligence and Promotion Planning and Optimization application. It contains the following sections:

- “Locating Host Specific Parameters” on page 7-1
- “Monitoring the PI and PPO Server” on page 7-1
- “Monitoring Promotion Calc Engine (PCE) Servers” on page 7-2
- “Monitoring Batch Jobs” on page 7-2
- “Model Migration” on page 7-6
- “Parallel Processing” on page 7-7
- “Other Maintenance Activities” on page 7-8

Locating Host Specific Parameters

The following files contain host specific settings. Many of these default to “localhost” but may be different in a production environment.

```
$installdir/config/SIT/sit-config.properties
$installdir/config/promote/promote.properties
$installdir/modules/tools/bin/storesetupdater.sh
$installdir/modules/tools/conf/jndi.properties
$installdir/modules/tools/conf/promote-cmdline.properties
$installdir/modules/pce/sample/deploy.sh
$installdir/modules/pce/sample/images/install_is.sh
$installdir/config/SIT/server-list.xml
```

Monitoring the PI and PPO Server

Besides the existing Oracle Application Server monitoring facilities for monitoring applications, the following URL can be used to provide a pulse check of the web component to ensure that it is running properly:

```
http://${WL_HOST}:${WL_PORT}/promote/service.do?command=status
```

If everything is running properly, this command will return the string “OK”

Monitoring Promotion Calc Engine (PCE) Servers

To monitor the PCE servers, execute the following Linux command on each PCE host:

```
bash$ pceserver.sh -query status
Querying PCE server status....
PCE server is up running.
```

If the PCE server is not responding, use the following command to retrieve the server load.

```
bash$ pceserver.sh -query load
Querying PCE server load....
PCE server is running 0 applications.
```

For a listing of additional PCE server tools, run the `pceserver.sh` command with no options. A list of available commands will be provided.

Monitoring Batch Jobs

The following section provides an introduction to batch jobs.

Overview

Batch operations are composed of tasks, steps, and agents.

- Tasks – tasks are aligned with the major batch operations. These must be scheduled using a management tool and typically are enabled on one host.
- Steps – each task is composed of a number of steps, such as waiting for a file, unzipping it, and processing it.
- Agents – tasks and steps are managed as a process outside the application server. Agents run within each application server.

Tasks

The following are the tasks currently configured in the system:

- `weekly.load` – this task is responsible for loading and processing the appropriate data passed to the application each week.
- `weekly.restate` – this task also runs weekly, but it is meant specifically for certain data files separate from `weekly.load`.
- `nightly.load` – running nightly, this task loads the data that needs to be processed on a nightly basis.
- `nightly.export` – this task also runs nightly and is responsible for preparing data for export back to the client.

Steps

Each task defines a list of steps required to execute the task. This list of steps is task specific and can be found within `$/AUTOMATION_BASE/config/<task>/Process.steps`.

The Job Log Table

Weekly and nightly feeds log their activities to the PR_MGMT_TASK table. The table below lists the columns in this table.

Table 7-1 PR_MGMT_TASK Table Attributes

Column Name	Data Type	Nullable	Data Default	Column ID	Primary Key	Comments
RUN_ID	Number (32,0)	No		1		
TASK_NAME	Varchar2 (40 byte)	No		2		
TASK_ID	Number (32,0)	Yes		3		
STEP_NAME	Varchar2 (80 byte)	No		4		
STEP_START	Date	No		5		
STEP_END	Date	Yes		6		
STEP_DURATION	Number (5.0)	Yes		7		
STEP_RESULT	Number (3,0)	No		8		
HOST	Varchar2 (200 byte)	No		9		
ARGUMENTS	Varchar2 (2048 byte)	Yes		10		
DESCRIPTION	Varchar2 (2048 byte)	Yes		11		

The application writes to this table with each step of its nightly or weekly batch cycle. The following is a brief example of what is written:

Table 7-2 Sample Nightly Batch Data within the PR_MGMT_TASK Table

RUN_ID	TASK_NAME	TASK_ID	STEP_NAME	STEP_RES	HOST	DESC
1	nightly.load		start	0	dev-app-101	Michaels Arts and Crafts: START in nightly.load (Run ID 20071018-1312) Automation log file is at ../mdc/operations/logs/automation/nightly.load.200710181312.log All the logs are at ../mdc/operations/logs
2	nightly.load	1	generic/wait.for.files.sh	0	dev-app-101	Running step 1/21 (generic/wait.for.files.sh) Step ended

Table 7–2 (Cont.) Sample Nightly Batch Data within the PR_MGMT_TASK Table

RUN_ID	TASK_NAME	TASK_ID	STEP_NAME	STEP_RES	HOST	DESC
3	nightly.load	1	update	0	dev-app-101	Beginning nightly processing...
...						
39	nighly.load	1	promote/load/data.sh	0	dev-app-101	Running step 20/21 (promote/load.data.sh) Step ended
40	nightly.load	1	update	0	dev-app-101	Nightly load complete
41	nightly.load	1	end	0	dev-app-101	Michaels Arts and Crafts: SUCCESS in nightly.load (Run ID 200710181312)

As this example shows, each task is made up of many steps linked together using the TASK_ID column. Each step receives one row in the above table.

Understanding Job Failures

The critical column to watch is STEP_RESULT. If the step succeeds, there will be a zero in this column. A non-zero value represents a failure.

Operators should be alerted when a step fails. The following table illustrates an example failure.

Table 7–3 Sample Task Failures within the PR_MGMT_TASK Table

RUN_ID	TASK_NAME	TASK_ID	STEP_NAME	STEP_RES	HOST	DESC
1	weekly.load		start	0	dev-app-101	Michaels Arts and Crafts: START in nightly.load (Run ID 20071024-1517) Automation log file is at ../mdc/operations/logs/automation/weekly.load.200710241517.log All the logs are at ../mdc/operations/logs

Table 7–3 (Cont.) Sample Task Failures within the PR_MGMT_TASK Table

RUN_ID	TASK_NAME	TASK_ID	STEP_NAME	STEP_RES	HOST	DESC
2	weekly.load	1	generic/ wait.for.files.sh	0	dev- app- 101	Running step 1/21 (generic/wait.for. files.sh) Step ended
3	weekly.load	1	fail	2	dev- app- 101	Michaels Arts and Crafts: FAILURE in weekly.load (RUN ID 200710241517)

The log file above demonstrates that the client failed to deliver the data files in the specified time window:

```
2007.10.24 15:17:17 Started processing weekly.load on 200710241517 with log at
../mdc/operations/logs/automation/weekly.load.200710241517.log.
2007.10.24 15:17:18 Running step generic/wait.for.files.sh (1/38):
2007.10.24 15:17:18 Step started at 2007.10.24 15:17:18.
2007.10.24 15:18:33 Have NOT received 'weekly.ash_lh_tbl.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.ash_mh_tbl.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.stage_mh_attrs_tbl.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_pr_like_merchandise.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_promo_offer_attr.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_promo_offer_merch.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_promo_offer.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_promo_store.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_promotions.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.wk_hist_sales_inv.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_mb_detail.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.data_ready.txt'.
2007.10.24 15:18:33 FATAL: Files are late and abort time has passed. Giving up.
2007.10.24 15:18:33 Step ended at 2007.10.24 15:18:33. Elapsed time is 75 seconds.
2007.10.24 15:18:34 Running procedure 'pr_mgmt.task_fail' with parameters '223, 2
, 'Demo Stores: FAILURE in weekly.load (Run ID 200710241517)''
2007.10.24 15:18:34 Ran procedure 'pr_mgmt.task_fail' successfully.
```

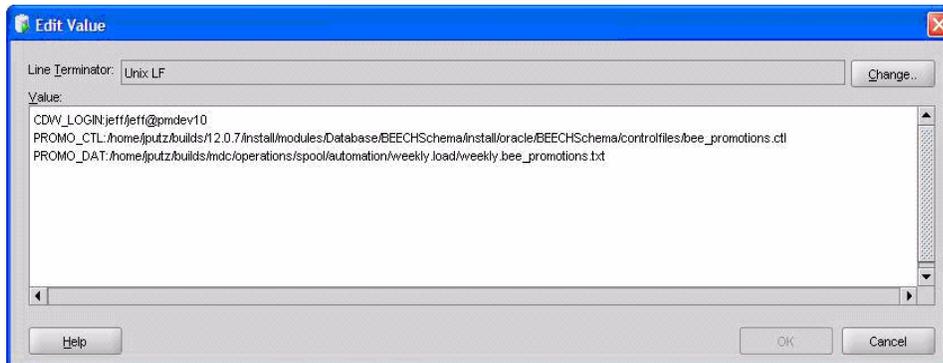
Locating Bad Data

A common failure is bad data received from the client. This can be manifested in several ways. Suppose the client mistakenly puts a header row on the input feed. The following table is a sample task log illustrating the failed step.

Table 7–4 Sample Bad Data Example

RUN_ID	TASK_NAME	TASK_ID	STEP_NAME	STEP_RES	HOST	DESC
239	weekly.load	226	start	2	dev- app- 101	Running step 13/38 (db/oracle/stage.file .sh) Step ended

Using SQL Developer, the file that failed can be located as follows:



The failed step was a result of a bad weekly.bee_promotions.txt file. Open the log file and scan it for the failed task as illustrated in the example below:

```
2007.10.24 16:14:35 Running step db/oracle/stage.file.sh (13/38):
2007.10.24 16:14:35 Step started at 2007.10.24 16:14:35.
2007.10.24 16:14:35 Staging logs at
'./mdc/operations/logs/database/stage/weekly.bee_
promotions.txt.200710241614.log'.
2007.10.24 16:14:35 Bad records file at
'./mdc/operations/logs/database/stage/weekly.bee_
promotions.txt.200710241614.bad'.
2007.10.24 16:14:36 Staging file
'./mdc/operations/spool/automation/weekly.load/weekly.bee_promotions.txt' with
control file
'./install/modules/Database/BEECHSchema/install/oracle/BEECHSchema/controlfiles/b
ee_promotions.ctl'...

Load completed - logical record count 23.
2007.10.24 16:14:36 FATAL: sqldr exited with an error.
2007.10.24 16:14:36 Step ended at 2007.10.24 16:14:36. Elapsed time is 1 seconds
```

The log files above demonstrate that the failure was a number format error in the actual offending record.

Restarting Jobs

Failed weekly batch jobs can be restarted using the \$AUTOMATION_BASE/scripts/do.weely.load script.

Jobs can also be restarted at a certain step by providing the line number of the job to the task script. For example, running \$AUTOMATION_BASE/scripts/do.weekly.load 7 would run weekly batch starting at step 7, assuming that the files have already been placed in their processing/spool directory.

Model Migration

To migrate a model that has been built from a development system to a production system do the following:

1. Capture the models to be migrated. Process them into intermediate files (.pmmlmv) using the script /install/modules/pce/bin/capture_model.sh.

2. Use a manual process to move the intermediate files from the source system to the target system.
3. Import the intermediate files in the target system and convert them into model files (.pmml). Deploy these migrated models using `/install/modules/pce/bin/import_model.sh`.

Parallel Processing

The two supported ways to run any process in parallel are:

- Manually run more than one process
- Run more than one process in operations

The Master Script ensures that the tasks identified as part of the configured SQL Templates are registered and executed only after the required pre-processing is complete. The Helper Process ensures that the master has registered the task. If not, it polls for the task in the cron process and the manual process exits without processing the task.

Manual Process

The following generic script acts as helper for the master:

```
process\scripts\common\process_exec_helper_manual_doit.sh.yapp.
```

This script is only responsible for executing the tasks in parallel. It checks the task table to identify if any process among the registered tasks is waiting for execution. If such a task is present, it executes the task or else terminates.

For example:

For lift models, the `1.build_lift_model_manual_helper.sh` resides under `\installdir\modules\pce\bin\build_lift_model`. This script executes tasks in parallel with the Main Process. This ensures that only Master Scripts are responsible for registering the list of the tasks for that process and that helpers are only responsible for selecting the task each time.

Running More Than One Process in Operations

The following script should be used when a task is executed as a cron job. The cron job for the helper process should be scheduled to execute a few minutes before the master cron job. The following generic script is provided that acts as helper for the master:

```
process \scripts\common\process_exec_helper_doit.sh.yapp.
```

This script is only responsible for executing the task runs in parallel. It checks the task table to identify if any process is pending to execute the registered task for the specific process; if present, it executes them. Helper processes wake every few minutes to determine if any registered process can be selected.

This script is only responsible for executing the task runs in parallel. It checks the task table to identify if any process among the registered tasks is waiting for execution. If such a task is present, it is executed. The helper process wakes every few minutes to determine if any registered process can be selected.

For example:

The Tae process has a master and helper script:

Master: `\mdc\operations\automation\step.lib\promote\pce\tae.sh`

Helper (internally calls `process_exec_helper_doit.sh`):

`\mdc\operations\automation\step.lib\promote\pce\tae_helper_run.sh`

The operation configuration steps for the automation scripts `\mdc\operations\automation\config` are:

- Master – `weekly.load`
- Helper – `weekly.helper.load`

The master scripts for the model build are:

- Main product – `build_lift_model.sh`
`\installldir\modules\pce\bin\`
- MDC – `doit.sh`
`\mdc\scripts\6.build_lift_models\4.build_and_measure`

The master scripts for tae are:

- Main product – `tae_run.sh`
`\installldir\modules\pce\bin`
- MDC – `doit.sh`
`\mdc\scripts\4.analyze_history\2.tae`

The updates to the SQL templates for customization are:

For the operations that Tae Runs uses:

`tae_run_sql_template.txt` residing under
`\mdc\operations\automation\step.lib\promote\pce\tae.`

For the operations MDC Tae Runs uses:

`tae_run_sql_template.txt` residing under `\mdc\scripts\4.analyze_history\2.tae.`

Any updates to this template affect the tae runs in the main MDC scripts.

Other Maintenance Activities

The following are additional actions that can be taken to ensure proper maintenance of the production environment.

Starting and Stopping the Server

Use the standard Oracle Application Server facilities to accomplish this.

Starting and Stopping the PCE

Use the following `pceserver.sh` commands to control and monitor the PCE server:

```
$bash pceserver.sh -start
$bass pceserver.sh -shutdown
```

Forecast Accuracy Indicator

This chapter explains the evaluation of a forecast for accuracy. It contains the following sections:

- [“Introduction” on page 8-1](#)
- [“Configuration” on page 8-1](#)
- [“Metrics” on page 8-2](#)

Introduction

The Forecast Accuracy Indicator is an enhancement to the PCE forecast prediction that evaluates the accuracy of a forecast by comparing current forecast data with historical data.

A rule-based decision tree based on a statistical analysis is used in the determination of the forecast accuracy. The decision tree is configured by Analytical Services (AS), using the `accuracy.properties` file. This file is used to configure the rules and the values used in the decision tree.

The UI displays the results of the accuracy determination.

Configuration

You can configure the forecast accuracy feature as follows:

Use the property `com.netperceptions.kde.rmi.server.RGIndicatorFlag=true` to enable or disable the Forecast Accuracy Indicator in the PCE.

For information about configuring the default thresholds that the UI uses to control the display of Red, Yellow and Green confidences, see the following properties in `promote.properties`:

- `promote.confidence.greenThreshold=70`
- `promote.confidence.yellowThreshold=30`
- `promote.confidence.redThreshold=0`

See the Merchandise Thresholds standard interface for information about the configuration of different thresholds for different areas of the merchandise hierarchy.

Metrics

This section lists the metrics used by AS to configure the accuracy.properties file. The supported rule operators in this file are:

=, <, >, <=, >=, !=

The metrics listed in [Table 8–1, " Model Metrics"](#) use the following abbreviations:

Abbreviation	Definition
XXX	attribute name
Metric String	abbreviation used in rule
CR	hard-coded constant expression
VI	metrics that are pre-evaluated as part of PCE start-up
X	data type
#	data type
Boolean	true/false

[Table 8–1, " Model Metrics"](#) contains the metrics used in accuracy.properties

Table 8–1 Model Metrics

Metric ID	Metric Description	Metric String	Abbreviation	Type	CR/VI=	Operator
PBL-1	Type of merchandise	PBL_MET.MERCH_TYPE		X	CR=[B/S]	=, !=
PBL-2	Maximum size of baseline window (# of weeks 5 or 9 for example)	PBL_MET.MAX_BL_PERIOD		#	CR=[?]	=, <, <=, >, >=, !=
PBL-3	Actual number of historic baseline weeks used for prediction	PBL_MET.TTL_GOOD_PERIODS		#	CR=[?]	=, <, <=, >, >=, !=
PBL-4	Number of dark weeks from all baseline window weeks	PBL_MET.DARK_PERIOD		#	CR=[?]	=, <, <=, >, >=, !=
PBL-5	Number of promotion weeks from historic baseline window	PBL_MET.PROMO_PERIOD		#	CR=[?]	=, <, <=, >, >=, !=
PBL-6	Number of clearance weeks from historic baseline window	PBL_MET.CLEARANCE_PERIOD		#	CR=[?]	=, <, <=, >, >=, !=
PBL-7	Number of gray weeks from historic baseline window	PBL_MET.GRAY_PERIOD		#	CR=[?]	=, <, <=, >, >=, !=
PBL-8	Average baseline sales of item during historic baseline window	PBL_MET.AVG_BL_SLS		##	CR=[?]	=, <, <=, >, >=, !=
PBL-9	Average baseline sales variance of item during historic baseline window	PBL_MET.AVG_BL_SLS_VAR		##	CR=[?]	=, <, <=, >, >=, !=
PBL-10	Future clearance indicator	n/a		n/a	n/a	n/a

Table 8–1 (Cont.) Model Metrics

Metric ID	Metric Description	Metric String Abbreviation	Type	CR/VI=	Operator
PBL-11	APC elasticity level of item	PBL_MET.PRICE_ELASTICITY_LEVEL	#	CR=[?]	=, <, <=, >, >=, !=
PBL-12	APC seasonality level of item	PBL_MET.SEAS_INDX_LEVEL	#	CR=[?]	=, <, <=, >, >=, !=
PBL-13	Seasonality Index of item	PBL_MET.SEAS_INDX	##	CR=[?]	=, <, <=, >, >=, !=
PBL-14	APC Price elasticity used for the item	PBL_MET.PRICE_ELASTICITY	##	CR=[?]	=, <, <=, >, >=, !=
PBL-15	Number of weeks between forecast as-of-date and ad-date	PBL_MET.WEEKS_TO_AD	#	CR=[?]	=, <, <=, >, >=, !=
MSC-1	Holiday ad?	n/a	n/a	n/a	n/a
MSC-2	Fiscal month of ad date	PBL_MET.FISCAL_MO	#	CR=[?]	=, <, <=, >, >=, !=
MSC-3	Fiscal quarter of ad date	PBL_MET.FISCAL_QUARTER	#	CR=[?]	=, <, <=, >, >=, !=
MSC-4	Error between Monkey Model forecast and PCE forecast.	n/a	n/a	n/a	n/a
MSC-5	Was like item	PBL_MET.LIKE_ITEM_USED_FLG	#	CR=[0/1]	=, !=
MSC-6	Was an offer level forecast used for predict baseline	PBL_MET.AGGR_PBL_USED_FLG	#	CR=[0/1]	=, !=
MDL-1	Lift model level of item	MDL_MET.MERCHANDISE_LEVEL	#	CR=[?]	=, <, <=, >, >=, !=
MDL-2	Lift model fitting error (MSE in pmml file)	MDL_MET.MSE	##	CR=[?]	=, <, <=, >, >=, !=
MDL-3	Lift model R2 (rsquare in pmml file)	MDL_MET.RSQUARE	##	CR=[?]	=, <, <=, >, >=, !=
MDL-4	Lift model intercept (intercept in pmml file)	MDL_MET.INTERCEPT	##	CR=[?]	=, <, <=, >, >=, !=
MDL-5	Lift model condition number (condNum in pmml file)	MDL_MET.CONDITION_NUMBER	##	CR=[?]	=, <, <=, >, >=, !=
MDL-6	Lift model F-statistic (fvalue in pmml file)	MDL_MET.FVALUE	##	CR=[?]	=, <, <=, >, >=, !=
MDL-7	Lift model p-statistic (pvalue in pmml file)	MDL_MET.PVALUE	##	CR=[?]	=, <, <=, >, >=, !=
MDL-8-1	Is current value < minimum historic value for each numerical predictor, where XXX is discount/price_ratio	MDL_MET.XXX.NUM_MIN	##	VI	=, <, <=, >, >=, !=
MDL-8-2	Is current value < maximum historic value for each numerical predictor	MDL_MET.XXX.NUM_MAX	##	VI	=, <, <=, >, >=, !=
MDL-8-3	STDDEV of current value from mean historic value for each numerical predictor	MDL_MET.XXX.NUM_STDDEV & MDL_MET.XXX.NUM_MEAN	##	VI	=, <, <=, >, >=, !=

Table 8–1 (Cont.) Model Metrics

Metric ID	Metric Description	Metric String Abbreviation	Type	CR/VI=	Operator
MDL-8-4	P-Value of predictor coefficient for each numerical predictor	MDL_MET.XXX.PVALUE	##	VI	=, <, <=, >, >=, !=
MDL-8-5	Standard error of predictor coefficient (sbk in pmml file) for each numerical predictor	MDL_MET.XXX.STD_ERROR	##	VI	=, <, <=, >, >=, !=
MDL-8-6	Variance inflation factor of predictor coefficient (vif in pmml file) for each numerical predictor	MDL_MET.XXX.VIF	##	VI	=, <, <=, >, >=, !=
MDL-8-7	The value of the predictor coefficient for each numerical predictor	MDL_MET.XXX.COEFFICIENT	##	VI	=, <, <=, >, >=, !=
MDL-9-1	Value of the predictor coefficient for categorical variables	MDL_MET.XXX.COEFFICIENT	##	VI	=, <, <=, >, >=, !=
MDL-9-2	P-value of the predictor coefficient for categorical variables	MDL_MET.XXX.PVALUE	##	VI	=, <, <=, >, >=, !=
MDL-9-3	Standard error of the predictor coefficient (sbk in pmml file) for categorical variables	MDL_MET.XXX.STD_ERROR	##	VI	=, <, <=, >, >=, !=
MDL-9-4	Variance inflation factor of the predictor coefficient (vif in pmml file) for categorical variables	MDL_MET.XXX.VIF	##	VI	=, <, <=, >, >=, !=

This chapter describes the reports that are available and provides details about configuring reports. It contains the following sections:

- “Introduction” on page 9-1
- “Available Reports” on page 9-1
- “Configuration Report Details” on page 9-3
- “Changing MicroStrategy Summary Levels” on page 9-4
- “MB Counts” on page 9-4

Introduction

Use the Standard Reports GUI to create and share new reports. All reports are based on a standard template. Several pre-defined reports are available, including reports that provide information on General Trends, Product Categories, and Individual Products.

Available Reports

PI-PPO provides the following reports. Because of rounding issues, the calculations in reports may be inaccurate. Metrics are calculated using full precision numbers; however, the reports only display two decimals. Validating these values manually using the metrics displayed in the reports can result in different results that are caused by the rounding. To prevent this, increase the number of decimals displayed in the reports.

- Affinity (Pull) – This report provides information about the affinity products or items that tend to sell well with other items. This report shows the affinity relationship over a longer period of time. This report contains metrics similar to the Affinity report, but also contains an additional column, “Pull Indicator”, that defines whether the relationship Likely, Unlikely, or Inconclusively drives sales between two items. It displays the affinity rules produced by the ARM application.
- Affinity Report – This report provides information about all affinity products and items that tend to sell with other items. It displays all of the affinity rules produced by the ARM application.
- Audit Trail Report – This report tracks changes made to a promotion at the user, date/time, and offer level. It also tracks changes to offers that affect the forecast, including Promotion dates, promotion phase changes, edits to vehicle types, added and deleted offers, offer status changes (submissions and approvals), and

any offer changes that affect the forecast (e.g. criteria, offer type, offer amount, demand drivers, forecast overrides, and position changes).

- Event Scorecard By Class Report – This report provides an analysis of the effect that individual classes have on the success of particular events. Viewers of this report also have the option of drilling into the metrics for Incr Allocated Non-Ad Sales, Incr Allocated Non-Ad GM, and Incr Allocated Non-Ad Units.
- Event Scorecard By Class/Offer Amount – This report provides an analysis of the effectiveness of different offer types and amounts. This report summarizes the offer type–amount performance within a class across multiple events. It enables a merchant to determine whether a %off discount has been more effective than a price point even if the effective discount was equivalent. Similarly, it can help determine whether a specific offer amount 25% or 30% off of a given offer type has been more effective historically. Viewers of this report also have the option of drilling into the metrics for Incr Allocated Non-Ad Sales, Incr Allocated Non-Ad GM, and Incr Allocated Non-Ad Units.
- Event Scorecard By Department/Offer Amount – This report provides an analysis of the effectiveness of different offer types and amounts. This report summarizes the offer type–amount performance within a department across multiple events. It enables a merchant to determine whether a %off discount has been more effective than a price point even if the effective discount was equivalent. Similarly, it can help determine whether a specific offer amount 25% or 30% off of a given offer type was more effective historically.
- Event Scorecard By Item Report – This report provides an analysis of the effect that individual items have on the success of particular events.
- Event Scorecard By Offer/Department Report – This report provides an analysis of the effect that each offer/department combination has on the success of particular events.
- Event Scorecard By Sub-Class Report – This report provides an analysis of the effect that individual Sub-classes have on the success of particular events. Viewers of this report also have the option of drilling into the metrics for Incr Allocated Non-Ad Sales, Incr Allocated Non-Ad GM, and Incr Allocated Non-Ad Units.
- Event Scorecard by Sub-Class/Offer Amount – This report provides an analysis of the effectiveness of different offer types and amounts. This report summarizes the offer type–amount performance within a sub-class across multiple events. It enables a merchant to determine whether a %off discount has been more effective than a price point even if the effective discount was equivalent. Similarly, it can help determine whether a specific offer amount 25% or 30% off of a given offer type has been more effective historically.
- Forecast Accuracy Report – This report compares the system and user (if one exists) predicted forecasts from a promotion created in Promotion Planning and Optimization against the sales results within Promotion Intelligence. Analysis is done only at the event level.
- Forecast Exception Report – This report provides information about changes in an offer's total forecast units. The changes in the forecast could be the result of system re-forecast process or a manual re-forecast by any user. The system has the ability to track forecast changes by units, sales or margin (one at a time).
- Model Accuracy Scorecard – This report provides information to help evaluate the efficiency of the predictive model.
- Overlapping SKUs Report – This report identifies cases where the same SKU exists in different offers in the same event. The specific offers and duplicate SKUs are

listed so that the user can correct the offers and avoid a pricing conflict in which the same SKU is promoted at different prices.

- TAE Assessment Report – This report provides information that can be used to evaluate the accuracy and completeness of the data generated by TAE.

Configuration Report Details

This section provides details about the two reports that are used during the configuration process. Details about the other reports can be found in the *Promotion Intelligence User Guide*.

Model Accuracy Scorecard Report

Use this report to evaluate the efficiency of the predictive model before you deploy it in the production environment. To test the model and produce this report, the PCE generates sales predictions for past ad events. Then, the application compares those predictions to the actual sales data gathered during the past ad event. The results are used to generate measures of error that can be used to evaluate the accuracy of the predictive model.

Report Prompts and Display

The report uses selected segments from the product hierarchy. The resulting report measures error only from the predictions that were generated for products that belong to the selected segment. This can be to evaluate how well a specific model predicts for a given segment. Results are grouped by the Model_ID that was used to produce the item/store level prediction.

Table 9–1 Model Accuracy Scorecard Report

Metric Number	Metric	Description
1	Model	The model's name from rdm
2	Focus Item	Focus Item (SKU Level)
3	Location	Location (Store Level)
4	Actual Quantity	Actual number of units sold
5	Predicted Quantity	Predicted number (generated by the model) of units sold
6	Chain Level Error	Mean Absolute Percent Error calculated from chain level aggregated units sold
7	Store Level Error	Total of Mean Absolute Percent Error calculated at the item/store level

TAE Assessment Report

Use this report to check the accuracy and completeness of TAE results. This report facilitates evaluating the data generated by the TAE process before integrating the data into the dataset. The report executes against the temporary table, PR_TAE_TEMP_METRIC.

Report Prompts and Display

The report prompts you to select the run ID, the merchandise hierarchy, and the ad event. Multiple data runs are identified by separate run IDs.

Table 9–2 TAE Assessment Report

Metric Number	Metric	Description
1	Run ID	Unique ID for TAE execution.
2	Focus Item	Item ID and description.
3	Promotion	Promotion description.
4	Ad Item MB Count	Count of ad market baskets that contain at least one focus item.
5	Item Baseline MB Count	Count of baseline market baskets that contain the item.
6	Index: Ad MBs to Baseline MBs	Count of item ad market baskets compared to the count of item baseline market baskets.
7	Status	OK. Bl_subst_code = 0 Substituted. Bl_subst_code = 1 No Result. Bl_subst_code in (2,3,4)
8	Substitute Item	Item used in substitution.

Changing MicroStrategy Summary Levels

PI-PPO reports use a default level (Department or MH level 4) of analysis. To change this level, do the following (demonstrated changing Summaries from Department (MH level 4) to Division (MH level 3):

1. Edit the Merchandise Level in Schema Objects/Attributes/Product Attributes
2. Select PI_ID and click Modify.

Summary Configurations

The PROMOTE_PROMO_OFFER_MH_SUMMARY parameter is a value for the INTERSECT_NAME in the ASH_CP_TBL standard interface. It specifies the level of aggregation for the merchandise hierarchy that is used to generate the totals for the scorecard by Offer/Department. The merchandise level should be the level that corresponds to the Department. The location level is not relevant to this aggregation.

The following summary configuration parameters specify the level of aggregation from the merchandise hierarchy that are used to generate the totals for the scorecard by merchandise hierarchy and offer amount.

INTERSECT_NAME	MERCHANDISE_LEVEL	LOCATION_LEVEL
PROMOTE_SCORECARD_MERCH_OFF_AMT_SUMM_3	DEPT	STORE
PROMOTE_SCORECARD_MERCH_OFF_AMT_SUMM_1	SUBCLASS	STORE
PROMOTE_SCORECARD_MERCH_OFF_AMT_SUMM_2	CLASS	STORE

MB Counts

The MB count is generated under the assumption that no overlap exists between promotions in the same event and that no overlap exists within events during the same calendar period. If this assumption is disregarded, double counting may occur when MB counts are done.

Internationalization

This chapter contains the following:

- [“Introduction” on page 10-1](#)
- [“Translation” on page 10-1](#)
- [“Supported Languages” on page 10-2](#)

Introduction

Internationalization is the process of creating software that can be translated more easily. Changes to the code are not specific to any particular market. The Promotion Intelligence and Promotion Planning and Optimization application has been internationalized to support multiple languages.

This chapter describes configuration settings and features of the software that ensure that the base application can handle multiple languages.

Translation

Translation is the process of interpreting and adapting text from one language into another. Although the code itself is not translated, components of the application that are translated include:

- Graphical user interface (GUI)
- Error messages

The following components are not translated:

- Documentation (Online Help, Release Notes, Installation Guide, User Guide, Operations Guide)
- Batch programs and messages
- Log files
- Configuration Tools
- Reports
- Demo data
- Training materials

Supported Languages

Promotion Intelligence and Promotion Planning and Optimization is available in the following languages:

- Chinese (Traditional)
- Chinese (Simplified)
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazilian)
- Spanish (Spain)

Promotion Intelligence and Promotion Planning and Optimization depend on both the browser settings and the regional settings to determine which language is being supported for a specific implementation.

The applications support a single language within a single installation. The applications do not support multiple currencies within a single installation.