Oracle® Retail Regular Price Optimization

Implementation Guide

Release 13.0.4

July 2009

ORACLE®

Oracle Retail Regular Price Optimization Implementation Guide, Release 13.0.4

Primary Author: Barrett Gaines

Licensing Note: This media pack includes a Restricted Use license for Oracle Retail Predictive Application Server (RPAS) - Enterprise Engine to support Oracle® Retail Regular Price Optimization only.

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

3 Build Scripts

4 Loading and Extracting Data

5 Integration

6 Configuration Considerations

7 Internationalization

List of Figures

List of Tables

Preface

The Oracle Retail Regular Price Optimization Implementation Guide describes post-installation tasks that need to be performed in order to bring Regular Price Optimization online and ready for production use.

Audience

This Implementation Guide is intended for the Regular Price Optimization application integrators and implementation staff, as well as the retailer's IT personnel. This guide is also intended for business analysts who are looking for information about processes and interfaces to validate the support for business scenarios within Regular Price Optimization and other systems across the enterprise.

Related Documents

For more information, see the following documents in the Oracle Retail Regular Price Optimization 13.0.4 documentation set:

- *Oracle Retail Regular Price Optimization Release Notes*
- *Oracle Retail Regular Price Optimization Installation Guide*
- *Oracle Retail Regular Price Optimization User Guide*

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, 13.0.2). 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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction

Oracle Retail Regular Price Optimization (RPO) assists retail price analysts in pricing hard-line and grocery items. It is suited for long lifecycle items with infrequent price changes. It recommends permanent prices and inventory volumes based on initial estimates of an item's total sales volume over a planning period, and on price-related sales of items and related items.

RPO includes grouping in its pricing analysis because it considers cross-item elasticities; that is, RPO considers how price changes for one item may affect the sales volume of other items that belong to the same demand group. Users can input objective functions and pricing constraints that define the optimization problem. Once these inputs are defined, the pricing optimizer recommends prices and associated volumes. What–if cases can also be created and evaluated side by side within the context of a pricing scenario. Once an analysis of the what–if cases and recommended prices is complete, the user can make a final decision to submit the recommended prices for the given set of merchandise items and locations.

For a more detailed overview of the functionality within RPO, see the *Oracle Retail Regular Price Optimization User Guide*.

Contents of this Guide

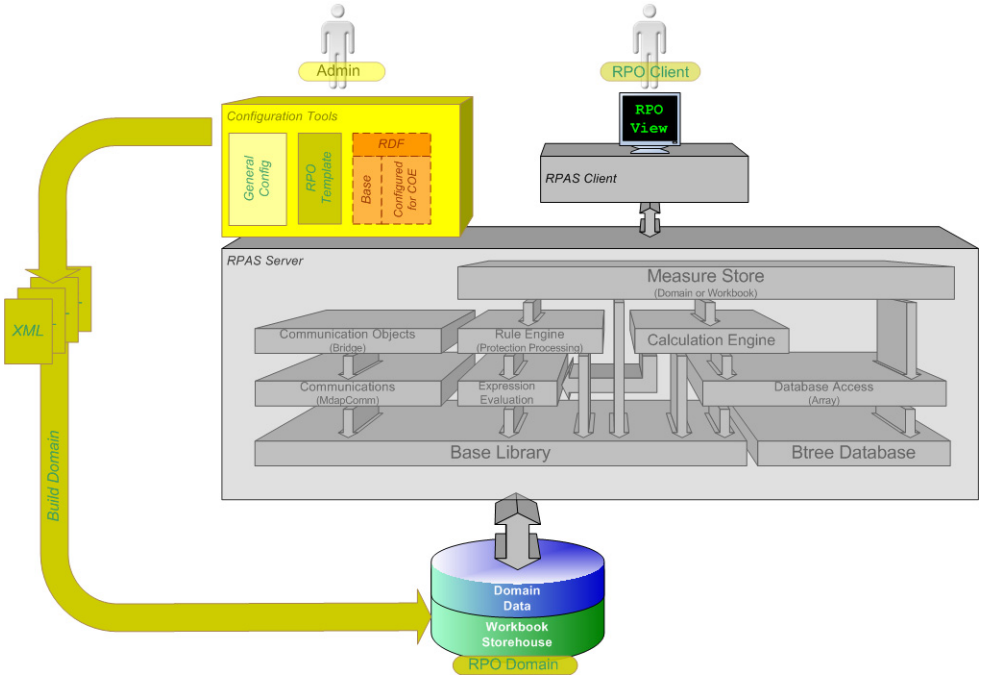
This implementation guide addresses the following topics:

- Chapter 1: Introduction. Overview of the RPO business workflow and skills needed for implementation.
- Chapter 2: Implementation Considerations. Explanation of the factors to take into consideration before performing the implementation.
- Chapter 3: Build Script. Information on building and patching the RPO - RPAS domain.
- Chapter 4: Loading and Extracting Data. Lists the measures that should be loaded to load functionally coherent set of data.
- Chapter 5: Integration. Overview of integration and explanation of the RPO data flow and integration script.
- Chapter 6: Configuration Considerations. Information on the functional changes or enhancements that can be made for RPO.
- Chapter 7: Internationalization. Translations provided for RPO.

RPO and the Oracle Retail Enterprise

Figure 1–1 shows the architecture of RPO and Oracle Retail Predictive Application Server (RPAS).

Figure 1–1 RPO and the Oracle Retail Enterprise



This diagram describes the RPO template applications. In the truest sense, RPO and other templates are not applications in the same way that the RPAS client is an application, since end users are not presented a user interface specific to the template. The RPO and other templates are predefined means to view specific types of data in the domain such that the RPAS client user interface is used to read and write to the domain.

The system administrator responsible for maintaining the RPAS Configuration Tools ensures that the appropriate templates are available. Each template has the following associated information to define its predefined attributes:

- Measures
- Special expressions
- Rules
- Workbook layout

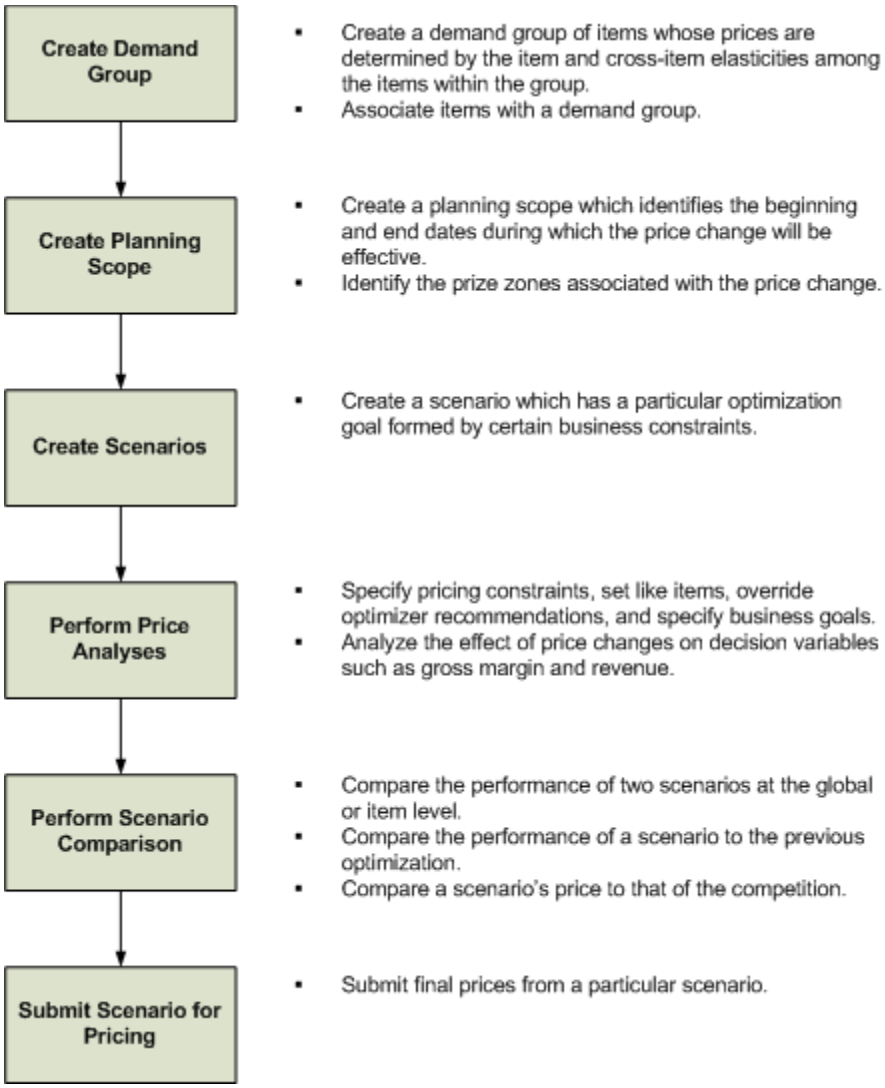
A client requests use of one of the templates by using the Configuration Tools. A number of XML files are then output which define the domain to be created. These XML files are used to build the specified domain incorporating all the attributes mentioned above that have been defined specifically.

Once the domain has been created, the end user can access the domain data through the RPAS client.

Business Process Workflow

Figure 1–2 shows a typical workflow for RPO.

Figure 1–2 RPO Workflow



Key Features of RPO

RPO provides the following features:

- Optimization of retail, grocery, and hard-line item regular prices for a defined time period subject to constraints
- Choice of optimization metrics: revenue, margin, highest item price, volume, Competitive Price Index (CPI), or any combination thereof
- Attention to global constraints on revenue, margin, and volume
- Attention to item and group price range and inter-item constraints
- Attention to discrete price ladder constraints
- Differentiation of prices across price zone or store level

Skills Needed for Implementation

The implementer needs an understanding of the following technical concepts:

- UNIX system administration, shell scripts, and job scheduling
- Performance constraints based on the retailer's infrastructure
- Technical architecture for RPO
- Retailer's hierarchical (SKU/store/day) data
- How to set up an RPAS domain
- A basic understanding of RPAS configuration and how to use the RPAS Configuration Tools
- Understanding of how RPAS rule language works
- Understanding of measures and dimension constructs
- Basic merchandising
- Basic forecasting
- The interaction of prices and volumes for items in assortment

2

Implementation Considerations

The following information needs to be considered before implementing RPO:

- Input Data
- Parametric Data
- Sizing Impacts
- Domain Partitioning
- Formatting
- Plug-ins
- Patch Considerations
- Batch Scheduling
- Implementation Phases
- Security
- Internationalization

Input Data

RPO uses the following required data:

- Base demand
- Current cost
- Current price

The following is optional data:

- Competition prices
- Price family data
- Unit of measure (UOM)
- Equivalent unit of measure (EUOM)
- Parametric data (see the Parametric Data section below)

Parametric Data

Item and cross-item elasticities quantify the effect of price on volume for both a single item and for other items whose demand is correlated with that item. Parametric data is not needed for Phase 1 implementation. Phase 2 requires item elasticities. Phase 3 requires both item and cross-item elasticities.

Sizing Impacts

The following factors can affect size requirements:

- SKU - Number of distinct items.

  Store - Number of physical, Web, and other distinct retail outlets.
- Calendar - Number of historical and future time periods in the domain. This impacts the overall size of the environment.
- Workbooks - Amount of space used by workbooks. This is typically greater than the domain itself. The number of workbooks is related to the number of users.

Domain Partitioning

Partitioning is done to avoid contention for resources. Building a workbook and committing data are two processes that can cause contention.

How data is partitioned has an impact on the business process. The RPO domain is defined as a global domain. For performance reasons, a simple domain is not recommended. There should be an even distribution of users across a set of local domains. For example, men's merchandise could be in a domain, women's merchandise in a domain, and children's merchandise in a domain. When a user is committing data in the men's merchandise domain, it does not affect the users in the women's or children's domains because of the use of partitioning.

Consider the following questions when defining the partitioning of the domain:

- How do I partition to meet my business needs?
- How do I partition my users?
- How do I create groups of users to further partition the solution?

Domain partitioning is supported on any dimension of the Product (PROD) hierarchy at or above Category. It is recommended that the RPO domain only be partitioned along the Product hierarchy. It is not recommended that the user partition along the Location (LOC) hierarchy, primarily because Price Zone will not be along the partitioned branch (it is an alternate, dynamic rollup) of the Location hierarchy, and all Price Zone level data would be stored in the Global Domain.

In RPO, all analytics are performed within the workbook and measures are not created during workbook creation. All DPM-enabled dimensions are Higher Base Intersections, and therefore there is a possibility of contention when too many DPM positions are created simultaneously.

Formatting

Formatting can be done in the configuration or the workbook after the domain is built.

- An implementer can create generic styles for the measures and assign them to measure components or realized measures. For each measure, these styles can be overridden on each workbook template. Formatting can only be changed by using the RPAS Configuration Tools. For more information, see the *Oracle Retail Predictive Application Server Configuration Tools User Guide*.
- Once the domain is built, the implementer can set up worksheet sizes and placements, exception value formatting, gridlines, and other formatting. The implementer instantiates a workbook of the template to set up specific formatting by using the Format menu. The updated format is then saved to the template so that it is available to all users for any newly created workbooks. For information on how to use the Format menu, see the *Oracle Retail Predictive Application Server User Guide*.

Plug-ins

Note: There is no Configuration Tools plug-in for RPO.

Plug-ins are application-specific Java code modules that run inside and automate the RPAS Configuration Tools to assist the implementer with specific application configuration. There are rules that an implementer must follow when configuring an application. A plug-in makes such adherence easier by automating parts of the configuration process and validity-checking entries that are made.

Patch Considerations

With a new release, there are two types of patches that can affect the RPO RPAS domain:

- Changes to the code in the RPAS libraries

  The configuration is not affected by this type of patch. For these types of changes, applying the patch is a straight forward process.
- Changes to the configuration

  These types of changes can be more complex. If a retailer has customizations in the configuration, the customizations must be redone on the new configuration before the patch is installed.

Patching Process

The RPO patch/upgrade process is not automated. As a result, any customizations achieved through configuration have to be manually reinstated for every configuration patch. Each RPO patch is accompanied with documentation that indicates the changes that were made so that the process of patching can be a focused effort.

Use the Configuration Tools Generate Reports utility to find differences between customized configurations and the released version of the configuration. The utility also finds the differences between patch or upgrade versions and the original released versions of the configuration. This facilitates manual patching of the specific customized solution.

Batch Scheduling

Batch scripts are lists of commands or jobs executed without human intervention. A batch window is the time frame in which the batch process must run. It is the upper limit on how long the batch can take.

RPO has no batch scripts that are central to optimization. Batch scripts are only needed for loading hierarchies and data files, as well as for exporting the recommended prices. The time that batch scripts are scheduled to run varies by implementation.

Implementation Phases

Many retailers want a way to manage rules for everyday pricing, while still maintaining the flexibility and foundation to layer on elasticity understanding and optimization as their organization and pricing practices mature. Some retailers just need to manage their rules (competitive rules, margin requirements, price families, inter-item rules) and do not need to consider elasticities or optimization. Other retailers would like to use elasticities to predict and understand volumes. Then, there are some retailers who would not only like to manage rules and consider elasticities but also perform optimization to recommend prices.

As a result of this business need, RPO can be implemented in three phases, allowing the user to benefit from the various degrees of complexity of RPO.

Phase 1

Phase 1 provides robust rules management capability. It does not consider price elasticities; it only satisfies the constraints set by the user. To use Phase 1, do not load any elasticity data.

Phase 2

Phase 2 provides robust rules management capability and what-if simulation which shows the item volume/financial impact based on the item elasticity. To use Phase 2, perform the following:

1. In the **Price Analysis** workbook, select the **Global Goals and Constraints** tab, and open the **Global Goals and Constraints** worksheet.
2. For the Goal measure, select **Highest Item Price** the goal option.
3. Click **Calculate.**
4. From the **File** menu, select **Commit Now**.

Selecting the Highest Item Price option as the goal measure ensures that only constraints are satisfied and no elasticities are used during the optimization. If any other option is selected for the goals measure, then the item price elasticity is used during the optimization run.

Phase 3

Phase 3 builds on the functionality of Phase 2, but it also allows consideration of cross-item elasticities in the what-if simulation and in the optimization. In a Phase 3 implementation, the revenue, gross margin, volume, CPI, and combination of global goals can be employed. And, by using cross-item elasticities, RPO captures cannibalization and halo effects among items.

Changing Phases

To upgrade RPO from one phase to the next, there is no need to patch the domain or upgrade any code. However, you must load the item elasticity data to move from Phase 1 to Phase 2. To move from Phase 2 to Phase 3, you must load the cross item elasticity data.

Security

To define workbook template security, the system administrator grants individual users, or user groups, access to specific workbook templates. Granting access to workbook templates provides users the ability to create, modify, save, and commit workbooks for the assigned workbook templates. Users are typically assigned to groups based on their user application (or solution) role. Users in the same group can be given access to workbook templates that belong to that group alone. Users can be assigned to more than one group and granted workbook template access without belonging to the user group that typically uses a specific workbook template. Workbook access is either denied, read-only, or full access. Read-only access allows a user to create a workbook for the template, but the user cannot edit any values or commit the workbook. The read-only workbook can be refreshed.

When users save workbooks, they assign one of three access permissions:

- World - Allow any user to open and edit the workbook.
- Group - Allow only those users in their same group to open and edit the workbooks.
- User - Allow no other users to open and edit the workbook.

Note: A user must have access to the workbook template in order to access the workbook, even if the workbook has world access rights.

For more information on security, see the *Oracle Retail Predictive Application Server Administration Guide*.

Internationalization

For more information on translation for RPO, see Chapter 7, "Internationalization".

3

Build Scripts

This chapter describes the setup that must be done before building the RPO - RPAS domain and the batch script that must be executed to build the domain.

Installation Dependencies

RPAS and RPO must be installed before setting up and configuring RPO.

- For information on installing RPAS, see the *Oracle Retail Predictive Application Server Installation Guide*.
- For information on installing RPO, see the *Oracle Retail Regular Price Optimization Installation Guide*.

Environment Setup

Before downloading the installation package to the UNIX server, a central directory structure to support the environment needs to be created. This central directory is referred to as `<rpo_directory>`. Set `<rpo_directory>` to the full path name to RPO home.

Note: The `$RPO_HOME` variable is not used.

RPAS Installation

The Java-based RPAS installation programs that are included with the installation package are used to install the server-side RPAS components on UNIX operating systems.

The RPAS installer performs the following functions:

- Installs the RPAS server
- Installs the Configuration Tools on the server

  On Windows, an InstallShield package is used to install the Configuration Tools.
- Defines the DomainDaemon port

RPO Installation

In addition to the RPAS installer, the installation package also includes the Java-based RPAS installation program for the RPO application.

The RPO installer performs the following functions:

- Downloads the configuration and batch scripts into the `<rpo_directory>/config` and `<rpo_directory>/bin` directories
- Downloads a set of sample hierarchy and data files into the `<rpo_directory>/input` directory
- Builds a sample domain at `<rpo_directory>/domain/RPO`

Custom Domain Build

To do a custom build of a domain:

1. Update the globaldomainconfig.xml file with the correct domain paths.
2. If needed, update the default environment variables in environment.sh.
3. Execute the buildRegPrice.sh script: ./buildRegPrice.sh

It is expected that the first time buildRegPrice.sh is executed, an error occurs when it tries to remove the old log file because a log file does not yet exist.

Environment Variables

In addition to the regular RPAS environment variables, including RPAS_HOME, you need to set up the following environment variables and export them:

- RPAS_JAVA_CLASSPATH
- RIDE_HOME
- PATH

Files Needed to Build the RPO RPAS Domain

Before the domain is built, the following types of files need to be set up:

- Standard RPAS hierarchy files
- RPO specific hierarchy files
- Data files

Standard RPAS Hierarchy Files

The following hierarchy files are needed:

- Calendar hierarchy file
- Product hierarchy file
- Location hierarchy file

Note: As with all standard RPAS hierarchies, these hierarchies are configurable. For information about configuring these hierarchies, see Chapter 6, "Configuration Considerations".

Calendar (CLND) Hierarchy File

File name: clnd.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–1 Calendar Hierarchy Fields

Field	Description
Day	Day ID
Day label	Day label
Week	Week ID
Week label	Week label
Mnth	Month ID
Mnth label	Month label
Qrtr	Quarter ID
Qrtr label	Quarter label
Year	Year number
Year label	Year label

Example:

```
20050130,01/30/2005,w01_2005,01/30/2005,JAN_2005,January
2005,Q1_2005,Quarter 1 2005,A2005,Year 2005
```

Product (PROD) Hierarchy File

File name: prod.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–2 Product Hierarchy Fields

Field	Description
Item	SKU ID
SKU label	SKU label
Clss	Class ID
Clss label	Class label
Scat	Sub Category ID
Scat label	Sub Category label
Cat	Category ID
Cat label	Category label
Dept	Department ID
Dept label	Department label
Chnl	Chain ID
Chnl label	Chain label

Example:

```
10426485,Bskt Lg Gld Wlw W/Red Vlv Lnr,217771339880,Cmas Storage
Baskets,21777133,Basket Collections,21777,Baskets &
Storage,217,Home Decor,0,Chain Top Level
```

Location (LOC) Hierarchy File

File name: loc.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–3 Location Hierarchy Fields

Field	Description
Str	Store ID
Str label	Store label
Zone	Zone ID
Zone label	Zone label
Regn	Region ID
Regn label	Region label
Cnty	Country ID
Cnty label	Country label
Chnl	Chain ID
Chnl label	Chain label
Entp	Enterprise ID
Entp label	Enterprise label
Przn	Price Zone ID
Przn label	Price Zone label

Example:

```
2068,Atl-Cumming,8,Atlanta GA DMA,H2,Hobby Mart,1,US,1,retailer
chnl ,0,retailer,9,Price Zone 9
```

RPO Hierarchy Files

The following are required hierarchy files needed for RPO:

- RHS Product
- Competition Product
- Price Ladder
- Price Point
- Planning Scope
- Item Group
- Demand Group
- Inter-item Constraints
- Inter-item Group Constraints
- Competition Constraints
- Competition Item Group Constraints
- Item Link Group
- What-if
- Scenario

RHS Product (PROR) Hierarchy File

RHS Product (PROR) hierarchy has the same format and content as Product (PROD) hierarchy. You should copy the Product hierarchy file and load it again for the RHS Product hierarchy.

File name: pror.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–4 RHS Product Hierarchy Fields

Field	Description
Item	SKU ID
SKU label	SKU label
Clss	Class ID
Clss label	Class label
Scat	Sub Category ID
Scat label	Sub Category label
Cat	Category ID
Cat label	Category label
Dept	Department ID
Dept label	Department label
Chnl	Chain ID
Chnl label	Chain label

Example:

```
10426485,Bskt Lg Gld Wlw W/Red Vlv Lnr,217771339880,Cmas Storage
Baskets,21777133,Basket Collections,21777,Baskets &
Storage,217,Home Decor,0,Chain Top Level
```

Competition Product (COMP) Hierarchy File

File name: comp.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–5 Competition Product Hierarchy Fields

Field	Description
Itec	Competitor SKU ID
Itec label	Competitor SKU label
Scac	Sub-category ID
Scac label	Sub-category label
Catc	Category ID
Catc label	Category label
Cmpt	Competitor ID
Cmpt label	Competitor label

Example:

```
RET10427707,Green BOWL,Bowl,Bowl Sub Category,Storage,Storage
category,Department, Department Example,Retail,Retail Example
```

Price Ladder Hierarchy File

A price ladder is a collection of acceptable price points for an item. RPO only recommends a price that is a price point on the price ladder.

For example, if an item's price ladder consists of price points spaced at $2.00 increments from $10 to $20, then RPO will only recommend a price at one of those price points.

File name: pl.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–6 Price Ladder Hierarchy Fields

Field	Description
Pldr	Price Ladder ID
Pldr label	Price Ladder label

Example:

```
1,Exhaustive Price Ladder from 0.99 to 99.99
```

Price Point Hierarchy File

A price point is an acceptable price for an item. A price ladder is a collection of price points for an item. RPO only recommends a price that is a price point on the price ladder.

File name: pp.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–7 Price Point Hierarchy Fields

Field	Description
Ppnt	Price Point ID
Ppnt label	Price Point label

Example:

```
1,price point 001
2,price point 002
3,price point 003
…
99,price point 099
```

Planning Scope Hierarchy File

Planning scopes can be created online, but it can also be loaded via the RPAS standard utility. This hierarchy requires a planning scope position named "example." Therefore, you should include in this hierarchy's loading file "example, example" as the first line. See the Planning Scope Loading section in this chapter for details on how to load the planning scope.

File name: ps.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–8 Planning Scope Hierarchy Fields

Field	Description
Pcsp	Planning Scope ID
Pcsp label	Planning Scope label

Example:

```
example,example
Pl001,Example planning scope
```

Item Group Hierarchy File

Item groups can be created online or loaded via the RPAS standard utility. There are two major purposes for the Item Group in RPO. First, it allows the user to group all the items together to set up the constraints rather than setting them up one at a time. For example, if you have an item group called Brand X Soda, you can set group level constraints so that the price of Brand X soda is always less than Brand Y soda.

Secondly, it allows users to set up the price family constraint. For example, if a group is defined for all Brand X 12-ounce. sodas (diet, regular, caffeine free, cherry), you can designate a price family to that group so that all types of Brand X's 12-ounce soda will have the same price.

See the Item Group Loading section in this chapter for details on how to load the item group.

File name: ig.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–9 Item Group Hierarchy Fields

Field	Description
Igrp	Item Group ID
Igrp label	Item Group label

Example:

```
itgp001,Item Group Example
```

Demand Group Hierarchy File

Demand Group represents a group of items for which the optimization solver will run. For example, a beverage demand group could contain name brand and store brand items that are optimized together.

Demand groups can be created online or loaded via the RPAS standard utility. See the Demand Group Loading section in this chapter for more details on how to load the demand group.

File name: dg.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–10 Demand Group Hierarchy Fields

Field	Description
Dgrp	Demand Group ID
Dgrp label	Demand Group label

Example:

```
Demand001,Demand Group Example
```

Inter-item Constraints Hierarchy File

Inter-item Constraints hierarchy is used to assign a unique identifier to each individual inter item constraint. It can be loaded via the RPAS standard utility or can be created online, although it is rarely created online.

File name: iicn.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–11 Inter-item Constraints Hierarchy Fields

Field	Description
Iicn	Inter-item Constraints ID
Iicn label	Inter-item Constraints label

Example:

```
001,Inter item constraint 001
```

Inter-item Group Constraints Hierarchy File

Inter-item Group Constraints hierarchy is used to assign a unique identifier to each individual inter-item group constraint. It can be loaded via the RPAS standard utility or can be created online, although it is rarely created online.

File name: iig.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–12 Inter-item Group Constraints Hierarchy Fields

Field	Description
Iigc	Inter-item Group Constraint ID
Iigc label	Inter-item Group Constraint label

Example:

```
C001,Inter-item group constraint 001
```

Competition Constraints Hierarchy File

Competition Constraints hierarchy is a dynamic hierarchy which is used to assign a unique identifier to each individual competition constraint. It can be loaded via the RPAS standard utility or can be created online, although it is rarely created online.

File name: cmc.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–13 Competition Constraints Hierarchy Fields

Field	Description
Cmpc	Competition Constraint ID

Example:

```
C001,Item competition constraint 001
```

Competition Item Group Constraints Hierarchy File

Competition Item Group Constraints hierarchy is used to assign a unique identifier to each individual competition item group constraint. It can be loaded via the RPAS standard utility or can be created online, although it is rarely created online.

File name: cig.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–14 Competition Item Group Constraints Hierarchy Fields

Field	Description
Cigc	Competition Item Group Constraint ID
Cigc label	Competition Item Group Constraint label

Example:

```
C001,Competition item-group constraint 001
```

Item Link Group Hierarchy File

This hierarchy is necessary to define the item link group. For more information about how the items link to other items, see the Item Link Group Mapping File section in this chapter.

File name: ilg.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–15 Item Link Group Hierarchy Fields

Field	Description
Ilg	Item Link Group ID
Ilg label	Item Link Group label

Example:

```
Ilg001,Item Link Group 001
```

What-if Hierarchy File

What-if hierarchy is a dynamic hierarchy which can be created online or loaded via the RPAS standard utility. This hierarchy requires the following three positions: "original," "recommend," and "user." Therefore, you should include these positions in the first three lines of this hierarchy's loading file:

- Original, Original
- Recommended, Recommended
- User, User

File name: wi.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–16 What-if Hierarchy Fields

Field	Description
Wtif	What-if ID
Wtif	What-if label

Example:

```
Original, Original
Recommended, Recommended
User, User
Wtif001, what if example
```

Scenario Hierarchy File

Scenario hierarchy is a dynamic hierarchy which can be loaded via the RPAS standard utility, but is usually created online. This hierarchy requires an "example" position. Therefore, you should include "example, example" in the first line of hierarchy file.

File name: scn.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–17 Scenario Hierarchy Fields

Field	Description
Scnr	Scenario ID
Scnr label	Scenario label

Example:

```
example,example,
Max_margin,max margin scenario
```

Required Data Files

The following data files are required:

- Base Demand
- Current Item Cost
- Current Item Price
- Price Ladder Value
- Item Default Price Ladder Assignment

Base Demand File

The demand forecast for the item at this specific location and week. It is assumed that this forecast already includes the seasonality factor. The location dimension is dependent on the optimization level, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: dl1itbdsp.csv.rpl
File format: comma-separated values file
Intersection: item/*location*/week

The following table describes the fields in the file:

Table 3–18 Base Demand Fields

Field	Description
Week	Week ID
Item	Item ID
Location	Optimization level's location dimension
Value	The forecast value

Example:

```
w01_2006,10033240,6,8.50
```

Current Item Cost File

The current cost of the item at that location. The location dimension is dependent on the optimization level, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: dl1orgitcst.csv.rpl
File format: comma-separated values file
Intersection: item/*location*

The following table describes the fields in the file:

Table 3–19 Current Item Cost Fields

Field	Description
Item	Item ID
Location	Optimization level's location dimension
Cost Value	Item cost at a certain location

Example:

```
10033240,1,3.17
```

Current Item Price File

The current price of the item at that location. The location dimension is dependent on the optimization level the user chooses, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: ol1orgitpc.csv.rpl
File format: comma-separated values file
Intersection: item/*location*

The following table describes the fields in the file:

Table 3–20 Current Item Price Fields

Field	Description
Item	Item ID
Location	Optimization level's location dimension
Price Value	Item price at a certain location

Example:

```
10033240,1,4.12
```

Price Ladder Value File

The price family to which the item belongs. The price family is a merchandise attribute. If the item does not belong to a price family, leave the field empty.

File name: pldrppt.csv.rpl
File format: comma-separated values file
Intersection: price ladder/price point

The following table describes the fields in the file:

Table 3–21 Price Ladder Value Fields

Field	Description
Pldr	Price Ladder ID
Ppnt	Price Point ID
Value	The price value

Example:

```
1,1,0.49
2,1,0.59
3,1,0.69
```

Item Default Price Ladder Assignment File

The item's default price ladder used when pricing for a specified location. The location dimension is dependent on the optimization level the user chooses, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: ol1itpcldr.csv.rpl
File format: comma-separated values file
Intersection: item/*location*

The following table describes the fields in the file:

Table 3–22 Item Default Price Ladder Assignment Fields

Field	Description
Item	Item ID
Location	Optimization level's location dimension
Value	The default Price Ladder ID

Example:

```
21777131,1,1
21777131,2,1
```

Optional Data Files

These data files are optional:

- Anchor Item Price
- Anchor Item Price Used
- Price Family
- Unit of Measures
- Equivalent Unit of Measures
- Competitor Current Price
- Competitor Last Price Check Date
- Competitor Price Type
- Competitor to Retailer Item Linkage
- Item Link Group Mapping
- Category/Store Price Zone Mapping

Anchor Item Price File

The anchor price of the item at that location. The location dimension is dependent on the optimization level the user chooses, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: ol1itancpc.ovr
File format: comma-separated values file
Intersection: item/*location*

The following table describes the fields in the file:

Table 3–23 Anchor Item Price Fields

Field	Description
Item	Item ID
Location	Optimization level's location dimension
Anchor Price Value	Item anchor price value at a certain location

Example:

```
10033240,1,4.12
```

Anchor Item Price Used File

The anchor price used measure indicates if the item will use the anchor price at that location. The location dimension is dependent on the optimization level the user chooses, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: ol1itancpcb.ovr
File format: comma-separated values file
Intersection: item/*location*

The following table describes the fields in the file:

Table 3–24 Anchor Item Price Used Fields

Field	Description
Item	Item ID
Location	Optimization level's location dimension
Value	"T" or "F"

Example:

```
10033240,1,T
```

Price Family File

The price family to which the item belongs. The price family is a merchandise attribute. If the item does not belong to a price family, leave the field empty. RPO can create an item group based on this file, but the user also can create the item group manually.

File name: ol1pcfml.csv.rpl
File format: comma-separated values file
Intersection: item

The following table describes the fields in the file:

Table 3–25 Price Family Fields

Field	Description
Item	Item ID
Price Family	The item family that the item belongs to

Example:

```
10213430 ,
10412409,1
```

Unit of Measure File

Unit of Measures (UOM) describes the unit of measure of the item (such as 6-pack, 12-pack, case, etc.). The UOM is not a required attribute for every item. It is useful when the user needs to make a distinction between several pack sizes of the same product. For example, Brand X packages its canned soda in 6-packs and 12-packs. The UOMs for those two products are 6.0 and 12.0, respectively.

This file is necessary if the user needs to set up inter-item constraints at UOM type. The unit price of 6-pack soda should be greater than the unit price of the 12-pack.

File name: ol1ituom.csv.rpl
File format: comma-separated values file
Intersection: item

The following table describes the fields in the file:

Table 3–26 Unit of Measure Fields

Field	Description
Item	Item ID
UOM Value	The sellable unit packsize

Example:

```
10213430 , 6.0
```

Equivalent Unit of Measures File

Equivalent Unit of Measures (EUOM) describes the multiple of a standard unit of measure (such as ounces, pounds, liters) contained in one unit. Like UOM, EUOM is not a required attribute for every item. It is useful when the user needs to make a distinction between several pack sizes, pack weights, or volumes of the same product. For example, Brand X packages its bottled soda in 1- and 2- liter bottles. The EUOM of the 1-liter soda is 1.0. The EUOM of the 2-liter is 2.0.

This file is needed if the user needs to set up inter-item constraints at EUOM type. The unit price of the 1 liter soda should be greater than the unit price of the 2 liter.

File name: ol1itequiu.csv.rpl
File format: comma-separated values file
Intersection: item

The following table describes the fields in the file:

Table 3–27 Equivalent Unit of Measures Fields

Field	Description
Item	Item ID
EUOM Value	The multiple of a standard unit of measure (such as ounces, pounds, etc.) contained in one unit.

Example:

```
10213430 , 2.0
```

Competitor Current Price File

The competitor's current price of an item at that location. The location dimension is dependent on the optimization level the user chooses, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: ol1cmpcurpc.csv.rpl
File format: comma-separated values file
Intersection: itec/*location*

The following table describes the fields in the file:

Table 3–28 Competitor Current Price Fields

Field	Description
Itec	Competitor's Item ID
Location	Optimization level's location dimension
Price	Competitor's Item Price

Example:

```
Comp10359233,1,4.49
```

Competitor Last Price Check Date File

The date of the last time the competitor's item price was checked at that location. The location dimension is dependent on the optimization level the user chooses, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: ol1cmppcchkdt.csv.rpl
File format: comma-separated values file
Intersection: itec/*location*

The following table describes the fields in the file:

Table 3–29 Competitor Last Price Check Date Fields

Field	Description
Itec	Competitor's Item ID
Location	Optimization level's location dimension
Date	The date of the last time the competitor's item price was checked at that location

Example:

```
Comp10343969,1,20080212
```

Competitor Price Type File

The competitor's item price type, such as promotion price or regular price, at that location. The location dimension is dependent on the optimization level the user chooses, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: ol1cmppctp.csv.rpl
File format: comma-separated values file
Intersection: itec/*location*

The following table describes the fields in the file:

Table 3–30 Competitor Price Type Fields

Field	Description
Itec	Competitor's Item ID
Location	Optimization level's location dimension
Type	Competitor's item price type, such as promotional, regular, etc.

Example:

```
Comp_10343969,1,Promo Price
```

Competitor to Retailer Item Linkage File

The linkage between the retailer item and competitor item at that location. This file is needed if retailer needs to set up the competition constraints.

The location dimension is dependent on the optimization level the user chooses, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: ol1ilgcomp.csv.rpl
File format: comma-separated values file
Intersection: item/competitor item/*location*

The following table describes the fields in the file:

Table 3–31 Competitor to Retailer Item Linkage Fields

Field	Description
Item	Retailer's Item ID
Itec	Competitor's Item ID
Location	Optimization level's location dimension
Linkage Value	Set to "T" in order to link the items. The absence of a record means there is no link between the retailer and the item.

Example:

```
10398713,competitor_10398713,1,T
```

Item Link Group Mapping File

This measure defines the item and RHS (right hand side) item linkage in the specified item link group. The value is to indicate if the item and RHS item will link in the item link group.

File name: ol1ilgititrmp.csv.rpl
File format: comma-separated values file
Intersection: item/iter/ilg

The following table describes the fields in the file:

Table 3–32 Item Link Group Mapping Fields

Field	Description
Item	Item ID
Iter	RHS (right hand side) Item ID
Ilg	Item Link Group ID
Linkage Value	Set to "T" in order to link the item and the RHS item in the item link group. The absence of a record means there is no link.

Example:

```
10412332,10412330,ilg1,T
```

Category/Store Price Zone Mapping File

Defines how the store maps to the price zone with various categories. This is not needed if the store to price zone rollup does not depend on Product.

File name: pzdynhrmp.csv.rpl
File format: comma-separated values file
Intersection: cat/store

The following table describes the fields in the file:

Table 3–33 Category/Store Price Zone Mapping Fields

Field	Description
Cat	Category ID
Str	Store ID
Price Zone Value	Price Zone ID

Example:

```
Cat_21777,store_1536,pricezone_7
```

Loading Dynamic Hierarchies

The following section describes how to load the following hierarchies:

- Demand Group Loading
- Item Group Loading
- Planning Scope Loading

Demand Group Loading

The steps to load the demand group are as follows:

1. Load Demand Group Hierarchy File

See the Demand Group Hierarchy File section in this chapter for instructions.

2. Load Demand Group Description

Defines the demand group's description.

File name: dgdesc.csv.rpl
File format: comma-separated values file
Intersection: dgrp (Demand Group ID)

Table 3–34 Demand Group Description Fields

Field	Description
Dgrp	Demand Group ID
Description	Demand Group description

Example:

```
Dgrp1,description of dgrp1
```

3. Load Demand Group Optimization Level

Defines the demand group's optimization level. The optimization level specifies at which level the user will optimize the price, such as item/price zone. The optimization level is customizable and is defined in the configuration.

File name: optlvl.csv.rpl
File format: comma-separated values file
Intersection: dgrp (Demand Group ID)

Table 3–35 Demand Group Optimization Level Fields

Field	Description
Dgrp	Demand Group ID
Optlvl	Optimization level

Example:

```
Dgrp1,1
```

4. Load Demand Group/Demand Level

Defines the demand group's demand level. The demand level specifies which level the base demand cost will be loaded. The demand level must be the same as the optimization level. That is, if optimization level 1 is assigned to the demand group, then demand level 1 should be assigned to it as well.

File name: dmndlvl.csv.rpl
File format: comma-separated values file
Intersection: dgrp (Demand Group ID)

Table 3–36 Demand Group/Item Association Fields

Field	Description
Dgrp	Demand Group ID
Dmndlvl	Demand level

Example:

```
Dgrp1,1
```

5. Load Demand Group/Item Association

Defines whether the item belongs to the price zone.

File name: dgitmbr.csv.rpl
File format: comma-separated values file
Intersection: item/dgrp (Item, Demand Group ID)

Table 3–37 Demand Group/Item Association Fields

Field	Description
Item	Item ID
Dgrp	Demand Group ID
Value	Set to "T" in order to make the item belong to a price zone. The absence of a record means there is no association.

Example:

```
10033240, Dgrp1, T
```

Item Group Loading

The steps to load the item group are as follows:

1. Load Item Group Hierarchy File

See the Item Group Hierarchy File section in this chapter for instructions.

2. Load Item Group Description

Defines the item group's description.

File name: igdesc.csv.rpl
File format: comma-separated values file
Intersection: igrp (Item Group ID)

Table 3–38 Item Group Description Fields

Field	Description
Igrp	Item Group ID
Description	Item Group description

Example:

```
Igrp1,description of item group igrp1
```

3. Load Demand Group/Item Group Level

Defines the item group and demand group association. This association is used to range the data while building the workbook.

File name: igdgasmt.csv.rpl
File format: comma-separated values file
Intersection: igrp/dgrp (Item Group ID/Demand Group ID)

Table 3–39 Demand Group/Item Group Level Fields

Field	Description
Igrp	Item Group ID
Dgrp	Demand Group ID
Value	Set to "T" to associate the item group to the demand group. The absence of a record means there is no association.

Example:

```
Igrp1,Dgrp1,1
```

4. Load Item/Item Group Association

Defines the item group and item association. This association will be used to range the data while building the workbook.

File name: ol1igitasmt.csv.rpl
File format: comma-separated values file
Intersection: igrp/item (Item Group ID, Item ID)

Table 3–40 Item/Item Group Association Fields

Field	Description
Igrp	Item Group ID
Dgrp	Demand Group ID
Dmndlvl	Set to "T" to associate the item to the item group. The absence of a record means there is no association.

Example:

```
10033240,Igrp1, T
```

Planning Scope Loading

The steps to load the item group are as follows:

1. Load Planning Scope Hierarchy File

See the Planning Scope Hierarchy File section in this chapter for instructions.

2. Load Planning Scope Description

Defines the planning scope's description.

File name: pscpdesc.csv.rpl
File format: comma-separated values file
Intersection: pscp/dgrp (Planning Scope ID, Demand Group ID)

Table 3–41 Planning Scope Description Fields

Field	Description
Pscp	Planning Scope ID
Dgrp	Demand Group ID
Description	Planning Scope's description

Example:

```
Pscp001,dgrp1,description of planning scope
```

3. Load Planning Scope Begin Date

Defines the planning scope's begin date.

File name: pscpbgndt.csv.rpl
File format: comma-separated values file
Intersection: pscp/dgrp (Planning Scope ID, Demand Group ID)

Table 3–42 Planning Scope Begin Date Fields

Field	Description
Pscp	Planning Scope ID
Dgrp	Demand Group ID
Date	Planning Scope's begin date

Example:

```
Pscp001,dgrp1,20090705
```

4. Load Planning Scope End Date

Defines the planning scope's end date.

File name: pscpenddt.csv.rpl
File format: comma-separated values file
Intersection: pscp/dgrp (Planning Scope ID, Demand Group ID)

Table 3–43 Planning Scope End Date Fields

Field	Description
Pscp	Planning Scope ID
Dgrp	Demand Group ID
Date	Planning Scope's end date

Example:

```
Pscp001,dgrp1,20090705
```

5. Load Planning Scope/Demand Group Association

Defines the planning scope and the demand group association. It is used to associate the demand group and planning scope. Each demand group can have multiple planning scopes.

File name: dgpscpmbr.csv.rpl
File format: comma-separated values file
Intersection: pscp/dgrp (Planning Scope ID, Demand Group ID)

Table 3–44 Planning Scope/Demand Group Association Fields

Field	Description
Pscp	Planning Scope ID
Dgrp	Demand Group ID
Value	Set to "T" in order to associate the planning scope to a demand group. The absence of a record means there is no association.

Example:

```
Pscp001,dgrp1,T
```

6. Load Planning Scope/Price Zones Association

Defines how many price zones will be in the specified planning scope. Each planning scope may include multiple price zones. The association between the planning scope and demand group is used to filter the planning scope options in the RPAS wizard. The association between the planning scope and price zone is use to filter the location hierarchy when building the price workbook.

File name: dgpscpprznmbr.csv.rpl
File format: comma-separated values file
Intersection: pscp/dgrp (Planning Scope ID, Demand Group ID)

Table 3–45 Planning Scope/Price Zones Association Fields

Field	Description
Przn	Price Zone ID
Pscp	Planning Scope ID
Value	Set to "T" in order to associate the planning scope to a price zone. The absence of a record means there is no association.

Example:

```
Przn01,pscp1,T
```

Parametric Data

The parametric data includes item elasticities and cross-item elasticities. Parametric data is not needed for Phase 1 implementation. Phase 2 requires item elasticities. Phase 3 requires both item and cross-item elasticities.

Item and Cross-item Elasticities

Item elasticities represent the relationship of price to volume. For example, if an item has a higher price point, you can expect that it will have a lower volume. Cross-item elasticities also represent the relationship of price to volume, but for the volume of other items that are related to that item. For example, if the price of item A changes, this will affect the prices of items B, C, and D.

The location dimension is dependent on the optimization level the user chooses, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: ol1gammasp.csv.rpl
File format: comma-separated values file
Intersection: item/RHS item/*location*

Table 3–46 Item and Cross-item Elasticities Fields

Field	Description
Item	Retailer's Item ID
Iter	RHS (right hand side) Item ID
Location	Optimization level's location dimension
Value	Cross-item elasticity value

Example:

```
10033240,10320667,1,-0.0192
10033240,10320665,1,0.0104
10033240,10320664,1,-0.0051
10033240,10320661,1,0.0465
```

Output from RPO to Legacy Systems

RPAS has a standard utility to export the data to a CSV flat file.

Final Price Recommendation

The final price recommendation. The location dimension is dependent on the optimization level the user chooses, meaning that if the optimization level is item/price zone, then the location dimension is price zone.

File name: ol1fitpc.csv
File format: comma-separated values file
Intersection: item/week/*location*

Table 3–47 Item and Cross-item Elasticities Fields

Field	Description
Item	Retailer's Item ID
Week	Week ID
Location	Optimization level's location dimension
Price	Output price from RPO

Example:

```
10033240,week01,1,5.99
```

Building the RPO - RPAS Domain

The script used to build or patch the RPO RPAS domain is described in this section. The script is located in the `<rpo_directory>/batch` directory.

Batch Designs

This section contains detailed information on the buildRegPrice.sh build script:

Building a Domain

Script
buildRegPrice.sh

Usage
buildRegPrice.sh <options -cdilbo> <flags -tg>

Table 3–48 BuildRegPrice.sh Build Script

Argument	Allowed Values	Description
Options	c	Configuration directory Default is <rpo_directory>/config
	d	Domain path Default is <rpo_directory>/domain
	i	Input directory Default is <rpo_directory>/input
	l	Log directory Default is <rpo_directory>/logs
	b	Demand Level
	o	Optimization Level
Flags	t	Set this flag to make a test build
	g	Set the flag to use debug function libraries (no argument)

Notes

- The script uses the Configuration Tools rpasInstall utility to build a domain. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The script also uses the mace and loadmeasure RPAS utilities. See the *Oracle Retail Predictive Application Server Administration Guide* for details on these utilities.
- All hierarchy and measure files are placed in `<rpo_directory>/input`
- The script creates two initial users: adm and usr. The passwords are initially set to be the same as the user ID (adm, usr).

Loading and Extracting Data

Data is loaded into RPO using the standard RPAS approach. See the *Oracle Retail Predictive Application Server Administration Guide* for details on formatting the load data files and on the utilities that enable administrators to load data into RPAS.

For specific information about loading and extracting data in RPO, see Chapter 4, "Loading and Extracting Data".

4

Loading and Extracting Data

Data is loaded into RPO using the regular RPAS approach. See the RPAS Administration Guide for details on formatting load data files and on utilities that enable administrators to load data into RPAS.

It is recommended that CSV (Comma Separated Variable) format files be used to reduce the size of load files.

The following subsections list the measures that should be loaded to load functionally coherent set of data, such as demand data. These sections list the measure names and detail any referential integrity requirements.

Loading Demand Data

The following table lists all measures that must be loaded for every item/location to ensure that RPO can properly work with the ScanPro model.

Table 4–1 Demand Data Measures

Measure	Measure ID	Intersection
Base Demand	dl#itbdsp	Product / Demand Level Location
Cross item elasticity	ol#gammasp	Product / Optimization Level Location
Seasonality	ol#itssn	Product / Optimization Level Location / Calendar

Loading Prices and Costs

The following table lists all measures that must be loaded to load prices and costs for all item locations.

Table 4–2 Price and Cost Measures

Measure	Measure ID	Intersection
Current Cost	dl#orgitcst	Product / Demand Level Location
Current Price	ol#orgitpc	Product / Optimization Level Location
Anchor Price	ol#itancpc	Product / Optimization Level Location
Anchor Price Used	ol#itancpcb	Product / Optimization Level Location

Loading Attribute Data

The following table lists all measures that must be loaded as attributes of items at various optimization levels. These attributes are used in creating price-family constraint, and in creating inter-item constraints that apply on UOM (Unit of Measure) or Equivalent UOMs.

Table 4–3 Attribute Measures

Measure	Measure ID	Intersection
Price Family	ol#pcfml	Merchandise
Unit of Measure	ol#ituom	Merchandise
Equivalent Unit of Measure	ol#itequiu	Merchandise

Loading Price Ladders

Price ladders are series of ascending price points. Each item must have a price ladder assigned to it, or the optimizer will display an error message. Price ladders can be assigned at run time.

The following tables list the hierarchies and measures that should be loaded for loading price ladders.

Table 4–4 Price Ladder Hierarchies

Hierarchy	Hierarchy ID
Price Point	pp
Price Ladder	pl

Table 4–5 Price Ladder Measures

Measure	Measure ID	Intersection
Price Ladder	pldrppt	Price Point / Price Ladder
Price Ladder Assignment	ol#itpcldr	Product / Optimization Level Location

Loading Competitor Data

The following table lists all measures that must be loaded to load competitor data.

Table 4–6 Competitor Data Measures

Measure	Measure ID	Intersection
Competitor Current Price	ol#cmpcurpc	Bottom Level Competitor / Optimization Level Location
Competitor last price check date	ol#cmppcchk dt	Bottom Level Competitor / Optimization Level Location
Competitor price type	ol#cmppctp	Bottom Level Competitor / Optimization Level Location
Competitor to Retailer Item linkage	ol#ilgcomp	Product / Bottom Level Competitor / Optimization Level Location

Loading Item Link Group Definitions

The following tables list the hierarchy and measures that must be loaded to load Item Link Groups used to establish LHS and RHS item relationships in Item Group level inter-item constraints.

Table 4–7 Item Link Group Hierarchy

Hierarchy	Hierarchy ID
Item Link Group	ilgp

Table 4–8 Item Link Group Measure

Measure	Measure ID	Intersection
Item Link Group	ol#ilgititrmp	Merchandise / RHS Merchandise / Item Link Group

Loading Dynamic Clustering of Stores to Price Zones

Price Zone clustering must be loaded in RPO. These two measures are used to load the Store to Price Zone mapping. See the section on dynamic hierarchies in the *RPAS Configuration Guide* for information regarding the content of these measures.

Table 4–9 Price Zone Measures

Measure	Measure ID	Intersection
Price Zone Map	pzdynhrmp	Trigger Product / Store (Location)
Price Zone Label	pzdynhrlbl	Trigger Product / Store (Location)

Loading Demand Group Definitions

At least one demand group definition must be loaded when seeding the system. You can load the demand group hierarchy and not load any measures. If you do load measures, be sure that the optimization and demand level numbers match. That is, if optimization level 1 is assigned to the demand group, then demand level 1 should be assigned to it as well.

Table 4–10 Demand Group Hierarchy

Hierarchy	Hierarchy ID
Demand Group	dg

Table 4–11 Demand Group Measures

Measure	Measure ID	Intersection
Description	dgdesc	Demand Group
Optimization Level	optlvl	Demand Group
Demand Level	dmndlvl	Demand Group
Demand Group Items	dgitmbr	Bottom Level Product / Demand Group

Loading Item Group Definitions

Though not required, item group definitions can be loaded into RPO. Since item groups are owned by demand groups, referential integrity rules have to be followed when creating load files:

- Each loaded Item Group must be assigned to one and only one Demand Group, even though the measure structure allows many-to-one mapping between Demand Groups and Item Group.
- All items in a Item Group must also be in the assigned Demand Group.

Table 4–12 Item Group Hierarchy

Hierarchy	Hierarchy ID
Item Group	ig

Table 4–13 Item Group Measures

Measure	Measure ID	Intersection
Description	igdesc	Item Group
Demand Group - Item Group Assignment	igdgasmt	Item Group / Demand Group
Item Group - Item Assignment	ol#igitasmt	Product / Item Group

Loading Planning Scopes

Though not required, planning scopes may be loaded. A planning scope belongs to a Demand Group and therefore the following referential integrity rules must be maintained:

- Even though the structure of most measures allows for many-to-many mapping between Demand Groups and Planning Scopes, the mapping is one-to-many.
- The Begin Date must be earlier than the End Date.
- Price Zones assigned to the Planning Scope must exist for the merchandises in the Demand Group.

Table 4–14 Planning Scope Hierarchy

Hierarchy	Hierarchy ID
Planning Scope	ps

Table 4–15 Planning Scope Measures

Measure	Measure ID	Intersection
Description	pscpdesc	Demand Group / Planning Scope
Begin Date	pscpbgndt	Demand Group / Planning Scope
End Date	pscpenddt	Demand Group / Planning Scope
Demand Group - Planning Scope Assignment	dgpscpmbr	Demand Group / Planning Scope
Price Zone - Planning Scope Assignment	dgpscpprznmbr	Price Zone (Location) / Planning Scope

Extracting Price and Volume Recommendations

Only the following measures are meant for extraction from RPO.

Table 4–16 Price and Volume Measures

Measure	Measure ID	Intersection
Item Price Plan	ol#fitpc	Product / Optimization Level Location / Calendar

5

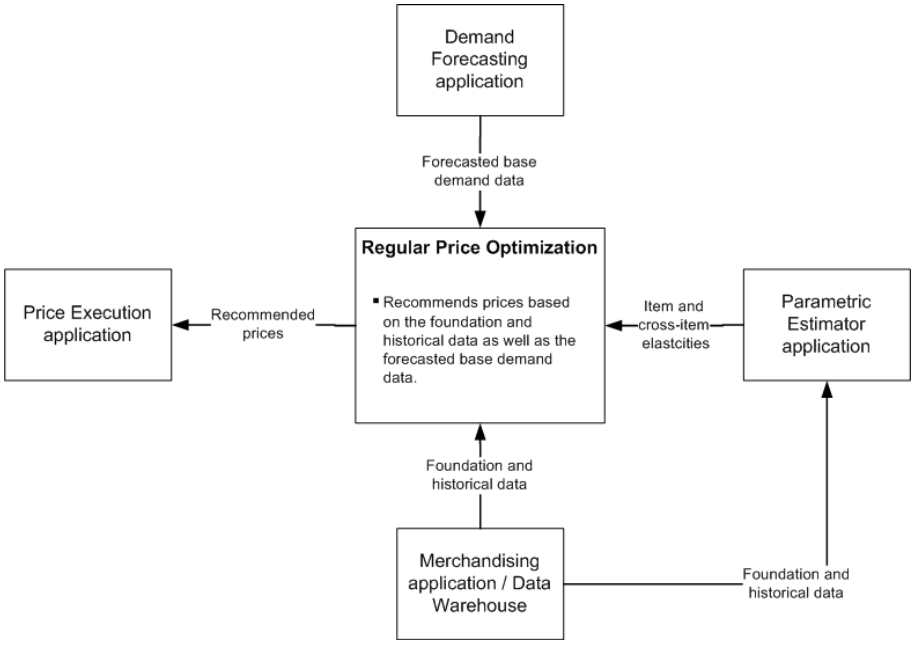
Integration

This chapter describes the interaction between RPO and other applications and the script used to load demand data.

Overview of RPO Data Flow

RPO is formally integrated only with Oracle Retail Demand Forecasting; although, it interacts with other applications and data warehouses as well in order to create optimized price recommendations. Figure 5–1 shows the interaction of RPO with other applications and the flow of data between the applications.

Figure 5–1 RPO Data Flow



Integration Interface Data Flow Description

These descriptions explain the each of the data flows in Figure 5–1.

From a Demand Forecasting Application (such as RDF) to RPO

- Sends forecasted base demand data, which includes level, trend, and seasonality of estimated demand. It does not, however, include price effects. The forecasted demand data needs to be time-phased, at item/price zone level.
- If using Oracle Retail Demand Forecasting, then RDF generates the base demand at the item-store level, and RPO aggregates the demand data from the item/store to the item/price zone level.

From a Parametric Estimator Application to RPO

- Sends item and cross-item elasticities. Item elasticity describes the relationship between an item's price and its volume. Cross-item elasticities also represent the relationship of price to volume but for the volume of other items that are related.

From a Merchandising Application and Data Warehouse to RPO

- Sends foundation and historical data such as item cost, item price, price ladder, item linkage.

From RPO to a Price Execution Application

- Sends recommended optimized prices.

Demand Data Load Script

Since RPO is not formally integrated with any application, no integration scripts exist. In order to load demand data, use the following load script:

```
RPO_load_demand_data.sh
```

This script loads forecast demand data at the item/store/week level. By using this specified intersection, the data may come from any forecasting system.

Use the following data format in the fixed width file:

Table 5–1 Demand Data

Data Entry	Description	Start	Width	Format	Example
Week	Date	1	8	YYYYMMDD	20091031
Item	Item ID	9	20	Item_ID	TZ748562A
Store	Store ID	29	20	Store_ID	0543
Value	The value of the forecasted demand of an item for a specified store during a specified week.	49	13	Any real number	27.40

Example:

```
20090101 TZ748562A            0543                27.40
```

6

Configuration Considerations

This chapter provides information on the functional changes or enhancements that can be made for RPO. It is recommended that implementers refrain from removing or changing any existing functionality beyond what is listed here. However, it is acceptable to add new measures and rules in such a way that there is no interference with the operation of existing rules.

Changing the Structure of Hierarchies

RPO has 17 hierarchies. Several of these hierarchies have been created to support abstract notions specific to RPO, and the structure of these hierarchies should not be changed (refer to Table 6–1); however, their load formats can be changed to suit the needs of the customer, except that the prefix length for their dimensions should always be zero (0). The structure of the other hierarchies can be changed but is subject to some conditions that are listed in the following subsections.

New hierarchies can be added for a particular implementation and interspersed between existing hierarchies. However, no existing hierarchies can be removed, nor should the order of existing hierarchies be changed. Additionally, the DPM (Dynamic Position Maintenance) status of any of the hierarchies should not be changed.

No changes should be made to dimensions and their configuration unless no other solution is possible. For example, there is no need to rename a dimension if only the label needs to be changed. Keeping the structure and names as standard as possible makes it easier to obtain product support when needed. Any changes that are made should be documented clearly, in case you need to obtain support.

Table 6–1 Non-Modified Hierarchies

Hierarchy Name	Hierarchy Label
PP	Price Point
PS	Planning Scope
IG	Item Group
DG	Demand Group
PL	Price Ladder
IICN	Inter Item Constraint
IIG	Inter-Item-Group Constraint
CMC	Competition Constraint
CIG	Competition Item Group Constraint
ILGP	Item Link Group
WI	What-if
SCN	Scenario

Calendar (CLND)

The Calendar hierarchy represents time in all RPAS solutions. It is a required hierarchy and must have a dimension named day (DAY). As it relates to RPO, the Calendar hierarchy is needed to store a time series of the seasonality component of the demand forecasts of items. These demand forecasts are used to calculate initial estimates of sales volume for all items.

Seasonality values can be stored at various levels along the Location and Product hierarchies; however, they are all stored along only one level of the Calendar hierarchy. RPO stores seasonality along the week (WEEK) dimension. A particular customer implementation might store seasonality along some other level on the Calendar hierarchy, but this level should represent the calendar and not notions such as week-of-the-month or week-of-the-year that aggregate non-sequential positions from the calendar.

If a customer implementation does store seasonality along a level other than week, the week dimension is not required in the Calendar hierarchy, and the base intersections of all measures in the RPO that utilize the week dimension must be changed to use this other dimension.

Other than abiding by these constraints, a customer implementation can structure the Calendar hierarchy in any way that best suits the customer's functional needs. Dimensions other than day and week have been included in RPO for the purpose of illustration. They can be modified or removed without requiring changes to any other elements of the configuration. Other dimensions and hierarchy branches can also be added without requiring changes to other elements of the configuration.

Note that none of RPO workbooks show the Calendar hierarchy in the worksheets.

Product (PROD)

The Product or Merchandise hierarchy represents the retailer's merchandise (that is, merchandise that the retailer retails through its retail channels). RPO does not enforce any constraints on the structuring of this hierarchy, but the configuration does use a few levels of this hierarchy extensively in workbook wizards, labeled intersections, rules, position queries, and measure values (Single Hier Select measures). Any changes to this hierarchy must be accompanied by changes to all these elements if they employ the particular level that is being modified or removed. Adding levels or branches or changing labels should not require any changes to the configuration.

From the hierarchy, the configuration employs Category (CAT), Class (CLSS), and Item (ITEM) in measure intersections, rules, and so on. To reduce patching and upgrade efforts, it is recommended that the names of these levels be left unchanged unless absolutely necessary.

Category is used in the wizard of the workbooks Analyst - Category Assignment and Demand Group Management. In Analyst - Category Assignment, it forms the level at which Product is assigned to Price Analysts for setting prices. Since these merchandise-analyst assignments are used by rules in the background to set up price hold constraints for the optimizer, Category is also used in rules to specifically set the level at which certain calculations occur.

In Demand Group Management, Category only serves to select Product at a reasonably high level for building the workbook; that is, it has no other functional significance and can be easily changed to something more suitable to the customer's needs without affecting the rest of the configuration.

Category is also used as the trigger dimension for clustering Store (STR) into Price Zone (PRZN). The dynamic hierarchy trigger dimension does not have to be the same as that used for Price Analyst assignment. The configuration is structured in this way for convenience. You can use two different levels for Analyst - Category Assignment and for the dynamic hierarchy trigger.

Class and Item are used in specification of the three optimization and demand levels in the configuration, and changing them requires several changes in the configuration. For more information, see the sections Changing Optimization Levels and RHS Product (PROR) in this chapter.

Note: When adding a new item to RPO, you must assign new parameters to the new item in the Like Item workbook and then assign it to a demand group in the Demand Group Management workbook. For more information, see the *Regular Price Optimization User Guide*.

RHS Product (PROR)

The RHS (right hand side) Product is intended to be a clone of the Product hierarchy. This hierarchy must have the same structure and the same positions as the Product hierarchy. Any changes to the Product hierarchy should be accompanied by the same changes to the RHS Product (PROR) hierarchy. This hierarchy is required to establish item linkage in inter-item pricing constraints which are specified with the use of item groups. It is also used to store cross-item elasticities when the ScanPro demand model is in use.

For ease of administration and to reduce the possibility of human error in loading hierarchy data, it is recommended that both the RHS Product and Product hierarchies be loaded in the same batch, from the same hierarchy load file.

Competition (COMP)

The Competition hierarchy represents the competitors' items and is required for loading competition prices and for specifying competition constraints. The structure of this hierarchy is flexible, with the exception of two levels that should not be modified: Competitor (CMPT) and Item (ITEC). The names of these levels are not hard-coded in the C++ implementation, meaning that the names can be changed, but all dependent configuration elements (measures, intersections, rules, and so on) would need to be modified to suit the change in names. To reduce patching and upgrading efforts, it is recommended that these names be left unchanged.

Competition constraints are specified using the Competitor (CMPT) level; therefore, that level should not be modified. Competition metrics, such as competitor prices, last price check date, and so on, are always loaded at the ITEC level, and therefore the ITEC level must always exist in this hierarchy. Such loading is independent of which optimization level is being used for a particular Demand Group (for instance, retailer items), and of the level at which they are priced (item, class, or category); thus, such loading should always link to the ITEC level in the Competition hierarchy. This linkage is necessary to avoid unnecessary complexity in preparing competition data load feeds.

Note: ITEC must always be the lowest level and CMPT must always be the highest level in this hierarchy.

All other levels in this hierarchy are optional and are available only as a convenience for the analyst examining competition metrics. These other levels have no functional correspondence to the Product hierarchy, nor to the levels within the Product hierarchy. Any perceived correspondence is loosely semantic or notional.

Note: The load format for this hierarchy can be modified at will.

Location (LOC)

The Location hierarchy represents the retailer's retail locations and their roll-ups. RPO imposes a few constraints on the structure of this hierarchy, but for the most part it is flexible.

The Location hierarchy must always have a level named PRZN (Price Zone). This hard-coded name is extensively used by the custom wizard code to filter data at various mappings in order to appropriately manage Planning Scopes. This must be a dynamic roll-up from the Store (STR) level, and has reasonable infrastructure built around this hierarchical structure. It is not advised to change this structure.

Other than STR and PRZN, all other levels are present for illustration purposes only and can be removed or modified as necessary.

Note: When you create a new price zone, it will not have any elasticity parameters until you load the parametric data from a parametric estimator application. For more information, see the *Regular Price Optimization User Guide*.

Changing Optimization Levels

An optimization level is the product/location level where price recommendations are made. Also known as a planning level, it is the lowest level that prices can be edited and viewed by a user. In RPO, every demand group can have a different optimization level associated with it. RPO comes preconfigured with three optimization levels: item/price zone, class/store, and class/price zone. This section discusses how one or more of these levels can be changed to be along a different intersection.

RPO has hundreds of measures at each optimization level, and the intersection of each must be updated if changes occur to the intersection of an optimization level. To make this task easier, RPO employs labeled intersections in the RPAS Configuration Tools. There are a few dozen labeled intersections that encompass all of the measures. By simply updating a labeled intersection, you can update all of the intersections of the measures that are encompassed in that labeled intersection. For example, if you wanted to change optimization level 1 from item/price zone to class/chain, then you would open the Configuration Tools interface, go to Labeled Intersections, and change the intersection specification of all labeled intersections that have the number '1' in their names to use the dimension CHNL instead of PRZN and to use CLSS instead of ITEM. This will automatically update the measure intersections.

Once the intersection specification is updated, perform the following:

1. Open the measure management tool and filter all measures that have names starting with Dl# and "Ol#" (# represents the demand level or optimization level).
2. Filter the Single Hier Select To column to those that have ITEM and change that column's value to CLSS.
3. Change the NA Value of the measure named Ol#IntNm to CLSSCHNL, where CLSSCHNL is the RPAS specification of the intersection. Each dimension's RPAS name is concatenated in any order. If the dimension name is fewer than four characters, it is padded with an underscore (_) to the right.
4. Update the Range attribute of the measure DmndLvl to reflect the selection of the new optimization level.

> **Note:** Making changes to the optimization level will require you to load the input data such as base demand, cost, price, and so on, at the same level.

5. Update the NA Value of the measure named Dl#IntNm to reflect the RPAS specification of the intersection.
6. Change the Range column for the measure name OptLvl to a value that helps you recognize the new level

All measure specifications are updated to account for the change in the optimization level.

Once measures have been updated, rules need to be updated to account for the change. Rule sets named PrcAnl and ScnAnl have several rule groups. Each optimization level has a subset of these rule groups, which can be identified by the fact that their name starts with "L#" (where # represents the optimization level). For example, if optimization level 1 is being changed, all rule groups that start with the name of L1 should be updated. These updates include changing dimension specification for specific expressions to match the optimization level. In these expressions, M# should be replaced with optimization level along the Product

hierarchy, and R# should be replaced with optimization level along the RHS Product hierarchy. L# should be replaced with optimization level along the Location hierarchy.

Changing Calculation of CPI to Retailer-As-Base

Competitive Price Index (CPI) is essentially a percent ratio that can be calculated either with the use of retailer revenue as base or competitor revenue as base. RPO uses competitor-as-base. To change RPO to use a retailer-as-base metric, services can change all CPI measure calculation rules for retailer-as-base.

Note: Detailed information about these rules is available in the *Oracle Retail Regular Price Optimization Release 13.0.4 Configuration Specification* document (Doc ID 843905.1), available at My Oracle Support (formerly Metalink):

For more information, contact Oracle Customer Support.

These changes need to be made in rule groups for the Price Analysis and Scenario Comparison workbooks.

Additionally, the optimizer input must be changed so that it works towards the CPI goal as described by this new metric. In RPO, user input is directly passed along to the optimizer. This should be changed by introducing an intermediate measure which is calculated as:

*Intermediate = (-100 * CPI Goal) / (CPI Goal + 100)*

This intermediate measure should be passed along to the optimizer instead of the original CPI goal. It is possible to make the denominator zero, and so it is advised to have a limit of -50 for the retailer-as-base CPI goal.

Switching Existing Measures to Cost-Based Accounting

In RPO, GM % (Gross Margin %) is the only measure that varies with the accounting method being used. In RPO, it is calculated based on retail accounting. This can be changed by changing the rules that calculate this metric.

Note: For more detail, refer to the *Oracle Retail Regular Price Optimization Release 13.0.4 Configuration Specification* document (Doc ID 843905.1), available at My Oracle Support (formerly Metalink):

For more information, contact Oracle Customer Support.

Adding New Analytic Measures

New measures can be added to any workbook to suit your needs. However, when adding measures and rules to calculate those measures, do not edit any of the existing rules; that is, you should not add expressions to existing rules and should not edit calculation of any measure in RPO (except those that have been explicitly mentioned in this guide).

In several cases it may be required to add measures that are calculated from other measures.

Note: For measure descriptions, refer to the *Oracle Retail Regular Price Optimization Release 13.0.4 Configuration Specification* document (Doc ID 843905.1), available at My Oracle Support (formerly Metalink):

For more information, contact Oracle Customer Support.

Adding Product Attributes

Attributes and dimension splitting can be used to aid in the creation of demand groups and item groups. Additionally, attributes can be used in a customer environment to help with price analysis by sorting or grouping (dimension splitting) items based on attributes. Any number of attributes can be added, and RPO does not impose any restriction on the addition of attributes. However, none of the existing attributes can be removed.

Changing the Workflow

None of the RPO workbooks can be removed from the solution except for the Optimizer Configuration workbook. The workbooks' access can be controlled in any way that suits your needs. The custom wizards, custom menus, and position queries of the workbooks cannot be changed. Additional workbooks and custom menus can be added; however, it should be ensured that these do not overlap in functionality with what already exists. Workbooks cannot be merged, and worksheets cannot be merged or removed.

7

Internationalization

Internationalization is the process of creating software that can be translated more easily. Changes to the code are not specific to any particular market. RPO has been internationalized to support multiple languages.

This section 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 may include the following:

- Graphical user interface (GUI)
- Error messages

The following components are not usually 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

The user interface for RPO has been translated into the following languages:

- German
- French
- Spanish
- Japanese
- Traditional Chinese
- Simplified Chinese
- Korean
- Brazilian Portuguese

- Russian
- Italian