

**Oracle® Retail Analytic Parameter Calculator for
Regular Price Optimization**

Implementation Guide

Release 13.3

January 2012

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Primary Author: Barrett Gaines

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.

Licensing Note: This media pack includes a Restricted Use license for Oracle Retail Predictive Application Server (RPAS) - Enterprise Engine to support Oracle® Retail Analytic Parameter Calculator for 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 **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.

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

(iii) 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.

(iv) 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.

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 |
| 1 Introduction | |
| Skills Required | 1-1 |
| Implementation Process | 1-2 |
| 2 Implementation Considerations | |
| Historical Data | 2-1 |
| Data Types | 2-1 |
| Historical Sales and Pricing | 2-2 |
| Dynamic Pricing Zone Mapping | 2-2 |
| Group of Substitutable Items | 2-2 |
| Partitioning | 2-3 |
| Formatting | 2-3 |
| Plug-ins | 2-3 |
| Patch Considerations | 2-3 |
| Batch Scheduling | 2-4 |
| Security | 2-4 |
| Internationalization | 2-4 |
| 3 Building the Domain | |
| Installation Dependencies | 3-1 |
| Environment Setup and Variables | 3-1 |
| RPAS Installation | 3-2 |
| RPAS Client Installation | 3-2 |
| APC-RPO Installation | 3-2 |
| Custom Domain Build | 3-2 |
| Files Required for Building the APC-RPO Domain | 3-3 |
| Standard RPAS Hierarchy Files | 3-3 |
| Calendar Hierarchy File | 3-3 |
| Merchandise Hierarchy File | 3-4 |
| Location Hierarchy File | 3-4 |
| APC-RPO Specific Hierarchy Files | 3-5 |

| | |
|--|-------------|
| Escalation Level Hierarchy File | 3-5 |
| Time Periods Hierarchy File..... | 3-6 |
| RHS Merchandise Hierarchy File | 3-6 |
| Data Files | 3-7 |
| Sales Units History..... | 3-7 |
| Sales Dollars History | 3-7 |
| Store to Price Zone Mapping..... | 3-8 |
| Ticket Price History | 3-8 |
| Minimum Ticket Price History | 3-9 |
| Maximum Ticket Price History | 3-9 |
| Promotion Indicator History (Optional)..... | 3-10 |
| Out-of-Stock Indicator (Optional) | 3-10 |
| Building the APC-RPO Domain..... | 3-11 |
| Configuration Files for the RPAS Fusion Client..... | 3-12 |
| Creating Users and User Groups..... | 3-12 |
| Loading and Extracting Data..... | 3-12 |

4 Data Flow

| | |
|---|------------|
| APC-RPO, RPO, RDF Integration..... | 4-1 |
| From APC-RPO to RPO..... | 4-1 |
| From APC-RPO to RDF..... | 4-2 |
| From RPO to RDF | 4-2 |
| From RDF to RPO | 4-2 |

5 ODI Integration

| | |
|---|------------|
| ODI Integration | 5-1 |
| Measure Data Integration | 5-1 |
| APC-RPO to RPO Package | 5-2 |
| Data Mapping for APC-RPO to RPO Package | 5-2 |
| APC-RPO to RDF Package..... | 5-2 |
| Data Mapping for APC-RPO to RDF Package..... | 5-2 |
| RDF to RPO Package | 5-2 |
| Data Mapping for RDF to RPO Package | 5-2 |
| RPO to RDF Package | 5-3 |
| Data Mapping for RPO to RDF Package | 5-3 |

6 Script Integration

| | |
|--|------------|
| Integration Scripts..... | 6-1 |
| apcrpo_export.sh..... | 6-1 |
| Outputs for RPO..... | 6-3 |
| Cross Item Elasticity | 6-3 |
| Anchor Price (Optional) | 6-4 |
| Minimum History Price (Optional) | 6-4 |
| Maximum History Price (Optional)..... | 6-5 |
| Maximum Price Increase Percentage Change..... | 6-5 |
| Maximum Price Decrease Percentage Change | 6-6 |

| | |
|---|------|
| Self Elasticity Standard Error | 6-6 |
| Self Elasticity T-Statistics..... | 6-7 |
| Outputs for RDF | 6-7 |
| Cross Item and Self Elasticity | 6-7 |
| Anchor Price | 6-8 |
| Price | 6-8 |
| 7 Configuration Considerations | |
| Calendar (CLND) Hierarchy | 7-1 |
| Product (PROD) Hierarchy..... | 7-2 |
| Location (LOC) Hierarchy..... | 7-2 |
| Escalation Levels (ELH) Hierarchy | 7-3 |
| Time Periods Hierarchy..... | 7-3 |
| RHS Product Hierarchy | 7-3 |
| 8 Batch Processing | |
| Scripts | 8-1 |
| apcrpo_batch.sh..... | 8-2 |
| apcrpo_batch_local.sh | 8-3 |
| buildAPCRPO.sh..... | 8-3 |
| loadApcRpoMeasures.sh | 8-4 |
| 9 Using the APC-RPO Plug-In | |
| About the APCRPO Parameters Window..... | 9-1 |
| Configuring the APC-RPO Solution | 9-2 |
| Configuring the Escalation Levels Tab | 9-2 |
| Generating Intersections, Measures, Rules, and Workbooks..... | 9-3 |
| 10 Internationalization | |
| Translation | 10-1 |
| Index | |

Send Us Your Comments

Oracle Retail Analytic Parameter Calculator for Regular Price Optimization
Implementation Guide, Release 13.3

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 Online Documentation available on the Oracle Technology Network Web site. 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 Analytic Parameter Calculator for Regular Price Optimization (APC-RPO) is a product used in conjunction with Oracle Retail Regular Price Optimization (RPO). APC-RPO generates values for price elasticities that are used by RPO to optimize item prices.

The *Oracle Retail Analytic Parameter Calculator for Regular Price Optimization Implementation Guide* describes the data requirements and batch processes that are required for implementation.

Audience

This Implementation Guide is intended for the following audiences:

- System administrators
- System analysts and programmers
- Integrators and implementation staff personnel

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit <http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

Related Documentation

For more information, see the following documents in the Oracle Retail Analytic Parameter Calculator for Regular Price Optimization Release 13.3 documentation set:

- *Oracle Retail Analytic Parameter Calculator for Regular Price Optimization Release Notes*
- *Oracle Retail Analytic Parameter Calculator for Regular Price Optimization Installation Guide*
- *Oracle Retail Analytic Parameter Calculator for Regular Price Optimization User Guide for the RPAS Classic Client*
- *Oracle Retail Analytic Parameter Calculator for Regular Price Optimization User Guide for the RPAS Fusion Client*

For more information on RPAS, refer to the Oracle Retail Predictive Application Server Release 13.3 documentation set.

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:
<https://support.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 re-create
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 13.3) or a later patch release (for example, 13.3.1). If you are installing the base release, additional patch, and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

Oracle Retail Documentation on the Oracle Technology Network

Documentation is packaged with each Oracle Retail product release. Oracle Retail product documentation is also available on the following Web site:

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

(Data Model documents are not available through Oracle Technology Network. These documents are packaged with released code, or you can obtain them through My Oracle Support.)

Documentation should be available on this Web site within a month after a product release.

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. |
| <code>monospace</code> | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

Introduction

The Analytic Parameter Calculator for Regular Price Optimization (APC-RPO) works in conjunction with Regular Price Optimization (RPO). It generates values for price elasticities, including self elasticities and halo and cannibalization cross elasticities. RPO uses this information to optimize item prices and reach desired goals such as gross margin and revenue.

The Implementation Guide addresses:

- APC-RPO configuration and the RPAS-specific considerations for the configuration.
- The setup required to build the APC-RPO domain.
- The scripts that must be executed to build the APC-RPO domain.
- The use of the APC-RPO plug-in.
- The scripts used to calculate the elasticities required by RPO.
- The scripts used to load measures and export measures.

Skills Required

In order to implement APC-RPO you must have an understanding of the following:

- UNIX system administration, shell scripts, and job scheduling.
- Performance constraints based on the retailer's infrastructure.
- Retailer's hierarchical (item/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.
- Analytical background, including an understanding of demand parameters, filter thresholds, results validation, price elasticities, cannibalization, and halo effects.

Implementation Process

The implementation of APC-RPO includes the following:

- The setup that must be done before the APC-RPO domain is built and the batch script that must be executed in order to build the domain.
- Using the plug-in to automate of the RPAS domain configuration to specify the partition dimension, dimensions, and hierarchies.
- Using the scripts provided to support APC-RPO, generate the demand parameters and export and load the generated measures.

Implementation Considerations

This chapter provides details regarding RPAS-specific implementation considerations for APC-RPO.

This chapter includes the following sections:

- [Historical Data](#)
- [Partitioning](#)
- [Formatting](#)
- [Plug-ins](#)
- [Patch Considerations](#)
- [Batch Scheduling](#)
- [Security](#)
- [Internationalization](#)

Historical Data

Five types of historical data are required for the regular price elasticity estimation. At a minimum, you must supply two years of continuous historical data; however, three years are recommended.

Data Types

The five types of historical data that are required include Historical Data and Pricing, Merchandise Hierarchy, Location Hierarchy, Promotion History, and Groups of Substitutable Items.

This chapter contains the descriptions for the required sets of data for:

- Historical Data and Pricing
- Promotion History
- Groups of Substitutable Items

[Chapter 3, "Building the Domain"](#) contains the descriptions for the required sets of data for:

- Merchandise Hierarchy
- Location Hierarchy

Note that the data file formats and the hierarchies must match. See [Chapter 3, "Building the Domain"](#) for examples of the data file formats.

Historical Sales and Pricing

At least two years of data, and preferably three years, are required at the item/location/week level. Note that a rolling window that is at least two years long is maintained by adding a new week of data and removing the oldest week.

Table 2–1 Data Element Descriptions

| Data Element | Description |
|--------------------------|---|
| Item | Item ID |
| Location | Location ID |
| Week | The calendar date that corresponds to the last day for a given week (for example, 2005-01-30) |
| Gross_Sales_Units | The net number of units sold (excluding returns) for the class at the location. |
| Gross_Sales_Amount | The sales amount (excluding returns) for the class at the location. |
| Ticket_Price | The item's ticket/sticker/shelf price (the most frequent of daily prices for a given item/location/week). The ticket price represents the price of a given item at a given location during a given day as displayed on the item's label, excluding any temporary or promotional price changes. |
| Min_Ticket_Price | The item's minimum ticket/sticker/shelf price (the lowest of daily prices for a given item/location/week). The ticket price represents the price of a given item at a given location during a given day as displayed on the item's label, excluding any temporary or promotional price changes. |
| Max_Ticket_Price | The item's maximum ticket/sticker/shelf price (the highest of daily prices for a given item at a given location during a given day as displayed on the item's label, excluding any temporary or promotional price changes. |
| Stockout_Flag (Optional) | The binary flag that indicates whether or not stockouts occurred for a given item/location/week. This flag is optional; it is used only if such historical data is available. |
| Promotion_Flag | The binary flag that indicates whether or not any promotions were active for a given item/location/week. |

Dynamic Pricing Zone Mapping

The Dynamic Price Zone Mapping feature provides flexibility so that a given item/store can be rolled up to a user-determined item/price zone. That is, the roll-up need not be straight up the hierarchy.

The input required for this is a measure data file called pz2prlocmap, which holds the mapping information. The format of the file is (item, store, price zone name). APC-RPO interprets this to mean that the (item, store) pair maps to the indicated price zone.

If this type of flexible roll-up is not necessary, then the file should only contain (item, store, price-zone-corresponding-to-store).

Group of Substitutable Items

The Substitutable Item Group is a configurable product level. When item hierarchies are designed, items should be grouped at some level as substitutable item groups.

Partitioning

Partitioning consists of splitting the data into separate sub-domains (under a global domain) according to the distinct members of the partition dimension. In APC-RPO, partitioning is done at the chain (the highest) level.

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

Formatting

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

- Each worksheet in the APC-RPO configuration has a measure order as well as measure styles that have been preconfigured. The measures can be displayed in the pre-configured order through the user interface. That format can then be saved to the template.

Using the RPAS Configuration Tools, 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. 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 edit formatting, see the *Oracle Retail Predictive Application Server User Guide for the Classic Client* or the *Oracle Retail Predictive Application Server User Guide for the Fusion Client*.

Plug-ins

Plug-ins are application-specific Java code modules that run inside and automate the RPAS Configuration Tools in order to assist the implementer with specific application configuration. An implementer must follow specific rules when configuring an application. A plug-in makes such adherence easier by automating parts of the configuration process and validating entries that are made. See [Chapter 9, "Using the APC-RPO Plug-In"](#) for information about the APC-RPO plug-in.

Patch Considerations

With a new release, there are two types of patches that can affect the APC-RPO 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 straightforward process.

- Changes to the configuration.

These types of changes can be more complex. If you have customizations in the configuration, you can use the `rpasConfigMgr` utility to determine the differences between your existing configuration and the new one. Then, you can use the utility to merge the two configurations. Any changes that cannot be applied are written to a change log. For more information, see the *Oracle Retail Predictive Application Server Configuration Tools User Guide*.

The script `buildAPCRPO.sh` can be used to apply the patch to the APC-RPO domain.

Batch Scheduling

Batch scripts are lists of commands or jobs executed without manual 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. Batch scripts are used for importing and exporting data. The retailer should decide the best time for running batch scripts within the available batch window.

Security

To define workbook template security, the system administrator grants individual users or user groups access to specific workbook templates. Users with access to workbook templates can 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 is not be able to edit any values or commit the workbook. The read-only workbook can be refreshed.

When users save a workbook, they assign one of three access permissions to the workbook:

- World – Allows any user to open and edit the workbook.
- Group – Allows only those users in their same group to open and edit the workbooks.
- User – Does not allow 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 for the Classic Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Fusion Client*.

Internationalization

For more information on translation for RPO, see [Chapter 10, "Internationalization"](#).

Building the Domain

This chapter describes the setup that must be done before the APC-RPO domain is built and the batch script that must be executed in order to build the domain.

This chapter contains the following sections:

- [Installation Dependencies](#)
- [Files Required for Building the APC-RPO Domain](#)
- [Building the APC-RPO Domain](#)
- [Configuration Files for the RPAS Fusion Client](#)
- [Creating Users and User Groups](#)
- [Loading and Extracting Data](#)

Installation Dependencies

The RPAS infrastructure (including the server and relevant client) and APC-RPO must be installed before APC-RPO can be set up and configured.

- For information on installing RPAS server and client, see the *Oracle Retail Predictive Application Server Installation Guide*.
- For information on installing APC-RPO, see the *Oracle Retail Analytic Parameter Calculator for Regular Price Optimization Installation Guide*.

Environment Setup and Variables

Before the installation package is downloaded to the UNIX server, a central directory structure to support the environment must be created. This central directory is referred to as <apcrpo_directory>. Set <apcrpo_directory> to the full path name to APC-RPO home.

Note: The \$APCRPO_HOME variable is not used.

In addition to the regular RPAS environment variables, including RPAS_HOME, you must export the following environment variable for all platforms:

```
export RPAS_JAVA_CLASSPATH="$RPAS_HOME/applib/aaijni.jar:$RPAS_
HOME/applib/aaiAPCRPO.jar:$RPAS_JAVA_CLASSPATH"
```

Note: There are additional Java environment variables that must be set for your particular operation system. These variables are the same for all applications on RPAS. See the "Java Environment" section of the *RPAS Installation Guide* for these environment variables.

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 the UNIX operating system.

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

RPAS Client Installation

The RPAS server installation package also includes the following RPAS clients:

- RPAS Classic Client: A Windows-based client interface for end-users and system administrators of an RPAS domain.
- RPAS Fusion Client: A Web-based client developed using Oracle Application Development Framework (ADF).

Each RPAS client installation package includes a separate installer to help you install the client. For more information on installing the RPAS clients, refer to the Oracle Retail Predictive Application Server Installation Guide.

APC-RPO Installation

The APC-RPO installer performs the following functions:

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

Custom Domain Build

To perform a custom build of a domain, perform the following steps:

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 buildAPCRPO.sh script:

```
./buildAPCRPO.sh
```

Files Required for Building the APC-RPO Domain

Before the domain can be built, the following types of files must be set up:

- [Standard RPAS Hierarchy Files](#)
- [APC-RPO Specific Hierarchy Files](#)
- [Data Files](#)

This section contains a description of each file and includes an example of each file.

Standard RPAS Hierarchy Files

The following hierarchy files are required:

- [Calendar Hierarchy File](#)
- [Merchandise Hierarchy File](#)
- [Location Hierarchy File](#)

Note: As with all standard RPAS hierarchies, these hierarchies are configurable as long as they adhere to the RPAS requirements on hierarchy structures.

For a description of each hierarchy structure, see [Chapter 7, "Configuration Considerations"](#).

Calendar Hierarchy File

File name: clnd.csv.dat

File format: comma-separated values file

[Table 3–1](#) describes the fields in this file.

Table 3–1 *Calendar Hierarchy Fields*

| Field | Description | Data Type | Importance |
|------------|---------------|-----------|------------|
| day | Day ID | Text | Required |
| day label | Day Label | Text | Required |
| week | Week ID | Text | Required |
| week label | Week Label | Text | Required |
| mnth | Month ID | Text | Required |
| mnth label | Month Label | Text | Required |
| qrtr | Quarter ID | Text | Required |
| qrtr label | Quarter Label | Text | Required |
| year | Year | Text | Required |
| year label | Year Label | Text | Required |

Example:

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

Merchandise Hierarchy File

File name: prod.csv.dat

File format: comma-separated values file

[Table 3–2](#) describes the fields in this file.

Table 3–2 Merchandise Hierarchy Fields

| Field | Description | Data Type | Importance |
|------------|--------------------|-----------|------------|
| item | item ID | Text | Required |
| item label | item label | Text | Required |
| Clss | Class ID | Text | Required |
| Clss label | Class label | Text | Required |
| Scat | Sub-category ID | Text | Required |
| Scat label | Sub-category label | Text | Required |
| Cat | Category ID | Text | Required |
| Cat label | Category label | Text | Required |
| Dept | Department ID | Text | Required |
| Dept label | Department label | Text | Required |
| Chn | Channel ID | Text | Required |
| Chn label | Channel label | Text | Required |

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 Hierarchy File

File name: loc.csv.dat

File format: comma-separated values file

[Table 3–3](#) describes the fields in this file.

Table 3–3 Location Hierarchy Fields

| Field | Description | Data Type | Importance |
|------------|------------------|-----------|------------|
| Stor | Store ID | Text | Required |
| Stor label | Store label | Text | Required |
| zone | Zone ID | Text | Required |
| zone label | Zone label | Text | Required |
| regn | Region ID | Text | Required |
| regn label | Region label | Text | Required |
| cnty | Country ID | Text | Required |
| cnty label | Country label | Text | Required |
| chnl | Chain ID | Text | Required |
| chnl label | Chain label | Text | Required |
| entp | Enterprise ID | Text | Required |
| entp label | Enterprise label | Text | Required |

Table 3–3 (Cont.) Location Hierarchy Fields

| Field | Description | Data Type | Importance |
|------------|------------------|-----------|------------|
| przn | Price zone ID | Text | Required |
| przn label | Price zone label | Text | Required |

Example:

```
2068,ATL-CUMMING,8,ATLANTA, GA DMA,H2,HOBBY LOBBY,1,US,1,retailer chnl
,0,retailer,9,Price Zone 9
```

APC-RPO Specific Hierarchy Files

The following are the hierarchy files specific to APC-RPO:

- [Escalation Level Hierarchy File](#)
- [Time Periods Hierarchy File](#)
- [RHS Merchandise Hierarchy File](#)

Escalation Level Hierarchy File

The Escalation Level Hierarchy groups the escalation levels together. It is a single dimension hierarchy.

File name: elh.csv.dat

File format: comma-separated values file

[Table 3–4](#) describes the fields in this file.

Table 3–4 Escalation Level Hierarchy Fields

| Field | Description | Data Type | Importance |
|------------|------------------------|-----------|------------|
| Elvl | Escalation Level | Text | Required |
| Elvl label | Escalation Level Label | Text | Required |

Example:

```
1,Escalation Level 1
```

Time Periods Hierarchy File

The Time Periods hierarchy is used internally to split the data points into two groups in order to calculate the sufficient statistic data.

File name: tprd.csv.dat

File format: comma-separated values file

Table 3–5 describes the fields in this file.

Table 3–5 Time Periods Hierarchy Fields

| Field | Description | Data Type | Importance |
|------------|-------------------|-----------|------------|
| tmpd | Time Period ID | Text | Required |
| tmpd label | Time Period Label | Text | Required |

Example:

```
1, time period 1
2, time period 2
```

RHS Merchandise Hierarchy File

The RHS Merchandise Hierarchy has the same format and content as the Merchandise Hierarchy. The file name is different.

File name: pror.csv.dat

File format: comma-separated values file

Table 3–6 describes the fields in this file.

Table 3–6 RHS Merchandise Hierarchy Fields

| Field | Description | Data Type | Importance |
|------------|--------------------|-----------|------------|
| item | item ID | Text | Required |
| item label | item label | Text | Required |
| Clss | Class ID | Text | Required |
| Clss label | Class label | Text | Required |
| Scat | Sub-category ID | Text | Required |
| Scat label | Sub-category label | Text | Required |
| Cat | Category ID | Text | Required |
| Cat label | Category label | Text | Required |
| Dept | Department ID | Text | Required |
| Dept label | Department label | Text | Required |
| Chn | Channel ID | Text | Required |
| Chn label | Channel label | Text | Required |

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
```

Data Files

This section describes the data files required by APC-RPO.

Sales Units History

Intersection: item/store/week

Description: The sales unit history.

File Name: pos.csv.ovr

[Table 3–7](#) describes the fields in this file.

Table 3–7 Sales Units History Fields

| Field | Description | Data Type | Importance |
|-------|----------------------|-----------|------------|
| week | Week ID | Text | Required |
| item | Item ID | Text | Required |
| store | Store ID | Text | Required |
| value | The sales unit value | Number | Required |

Example:

w01_2006,10033240,6,8.50

Sales Dollars History

Intersection: item/store/week

Description: The sales dollars history.

File Name: posdllr.csv.ovr

[Table 3–8](#) describes the fields in this file.

Table 3–8 Sales Dollars History Fields

| Field | Description | Data Type | Importance |
|--------------|-------------------------|-----------|------------|
| week | Week ID | Text | Required |
| item | Item ID | Text | Required |
| str | Store ID | Text | Required |
| sales amount | The sales dollar amount | Number | Required |

Example:

Week1,1234545,store1,401.49

Store to Price Zone Mapping

Intersection: cat/store

Description: Defines how a store maps to the price zone for a given merchandise category.

File Name: posdllr.csv.ovr

[Table 3–9](#) describes the fields in this file.

Table 3–9 Store to Price Zone Mapping Fields

| Field | Description | Data Type | Importance |
|---------------|---------------|-----------|------------|
| cat | Category ID | Text | Required |
| str | Store ID | Text | Required |
| price zone ID | Prize Zone ID | Text | Required |

Example:

```
cat123,store1,przn1
cat234,store1,przn2
```

Ticket Price History

Intersection: item/store/week

Description: The ticket price history.

File Name: price.csv.ovr

[Table 3–10](#) describes the fields in this file.

Table 3–10 Ticket Price History Fields

| Field | Description | Data Type | Importance |
|--------------|--------------|-----------|------------|
| week | Week ID | Text | Required |
| item | Item ID | Text | Required |
| str | Store ID | Text | Required |
| ticket price | Ticket Price | Number | Required |

Example:

```
Week1,1234545,store1,1.49
```

Minimum Ticket Price History

Intersection: item/store/week

Description: The minimum ticket price history.

File Name: minprice.csv.ovr

[Table 3–11](#) describes the fields in this file.

Table 3–11 Minimum Ticket Price History Fields

| Field | Description | Data Type | Importance |
|----------------------|----------------------|-----------|------------|
| week | Week ID | Text | Required |
| item | Item ID | Text | Required |
| str | Store ID | Text | Required |
| minimum ticket price | Minimum Ticket Price | Number | Required |

Example:

Week1,1234545,store1,1.49

Maximum Ticket Price History

Intersection: item/store/week

Description: The maximum ticket price history.

File Name: maxprice.csv.ovr

[Table 3–12](#) describes the fields in this file.

Table 3–12 Maximum Ticket Price History Fields

| Field | Description | Data Type | Importance |
|----------------------|----------------------|-----------|------------|
| week | Week ID | Text | Required |
| item | Item ID | Text | Required |
| str | Store ID | Text | Required |
| maximum ticket price | Maximum Ticket Price | Number | Required |

Example:

Week1,1234545,store1,1.49

Promotion Indicator History (Optional)

Intersection: item/store/week

Description: The promotion indicator history.

File Name: promoindcsv.ovr

Table 3–13 describes the fields in this file.

Table 3–13 Promotion Indicator History Fields

| Field | Description | Data Type | Importance |
|------------|--|-----------|------------|
| week | Week ID | Text | Required |
| item | Item ID | Text | Required |
| location | The location dimension for the optimization level | Text | Required |
| promo flag | A value of T indicates that the promotion occurred | Char | Required |

Example:

Week1,1234545,store1,T

Out-of-Stock Indicator (Optional)

Intersection: item/store/week

Description: The historical out-of-stock indicator.

File Name: outstckindcsv.ovr

Table 3–14 describes the fields in this file.

Table 3–14 Out-of-Stock Indicator Fields

| Field | Description | Data Type | Importance |
|-------------------|---|-----------|------------|
| week | Week ID | Text | Required |
| item | Item ID | Text | Required |
| location | The location dimension for the optimization level | Text | Required |
| out-of-stock flag | A value of T indicates that the out-of-stock occurred | Char | Required |

Example:

Week1,1234545,store1,T

Building the APC-RPO Domain

The buildAPCRPO.sh script used to build or to apply a patch to the APC-RPO domain is described in this section. The script is located in the <apcrpo_directory>/batch directory.

Script

buildAPCRPO.sh

Usage

buildAPCRPO.sh <options -cdil> <flags -gpt>

Table 3–15 describes the domain arguments for buildAPCRPO.sh.

Table 3–15 Domain Arguments for buildAPCRPO.sh

| Arguments | Allowed Values | Description |
|-----------|----------------|---|
| options | c | Configuration directory Default is <apcrpo_directory>/config |
| | d | Domain path Default is <apcrpo_directory>/domain |
| | i | Input directory Default is <apcrpo_directory>/input |
| | l | Log directory Default is <apcrpo_directory>/logs |
| flags | g | Set this flag to use debug function libraries |
| | p | Set this flag for a patch build |
| | t | Set this flag for a test build |

Notes

- The rules in the common_data group are crucial to the batch process and should never be modified by the retailer.
- The script uses the Configuration Tools rpaInstall utility to build a domain. See the *Oracle Retail Predictive Application Server Administration Guide for the Classic Client* or the *Oracle Retail Predictive Server Administration Guide for the Fusion Client* for details on this utility.
- The script also uses the following RPAS utilities: mace and loadmeasure. See the *Oracle Retail Predictive Application Server Administration Guide for the Classic Client* or the *Oracle Retail Predictive Server Administration Guide for the Fusion Client* for details on this utility.
- All hierarchy and measure files are placed in the <apcrpo_directory>/input directory.

Configuration Files for the RPAS Fusion Client

Use the APC-RPO installation software to install the activity task flow and online help files for the RPAS Fusion Client. The RPAS Fusion Client must be installed before the activity task flow files can be installed. For more information on installing the RPAS Fusion Client, refer to the *Oracle Retail Predictive Application Server Installation Guide*.

During the RPAS Fusion Client installation, the installer automatically sets up the RPAS domain connection configurations in the ProfileList.xml file. In case you choose to set up the domain connection after the installation or set up an additional domain, you must manually set up the connection. For more information, refer to the *Oracle Retail Predictive Application Server Administration Guide for the RPAS Fusion Client*.

Creating Users and User Groups

For greater security, users and user groups are not automatically created when you build or apply a patch to a domain. To create users and user groups, you must use the usermgr utility. For more information about usermgr, see the *Oracle Retail Predictive Application Server Administration Guide for the Classic Client* or the *Oracle Retail Predictive Server Administration Guide for the Fusion Client*.

Loading and Extracting Data

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

This chapter describes the data flow among APC-RPO, RPO, and RDF.

APC-RPO, RPO, RDF Integration

This section describes the integration and data flow between APC-RPO, RPO, and RDF.

Figure 4–1 APC-RPO, RPO, RDF Integration



From APC-RPO to RPO

- Sends item and cross-item elasticities of items. RPO uses these elasticities to optimize prices.
- Sends maximum and minimum historical prices of items. RPO uses this data to optimize prices.
- Sends anchor prices of items. Anchor prices are the baseline prices that APC-RPO uses to calculate price elasticity. RPO uses the anchor prices to calculate price drift metrics.

- Sends maximum price increase and decrease percentages, self-item elasticity standard errors, and self-item elasticities t-statistics. RPO uses the maximum price increase and decrease percentages to setup up the default minimum and maximum price percentage change. Meanwhile, the RPO user can refer to the self-item elasticity standard error and t-statistics to adjust the price constraint.

From APC-RPO to RDF

- Sends regular price item elasticities to RDF. These item elasticities, together with the price plan, allow RDF to calculate the item elasticity lift.
- Sends regular price cross-item elasticities to RDF. There are two types of cross-item elasticities: halo and cannibalization. These cross elasticities, together with the price plan, allow RDF to calculate the cross-item lift for both halo and cannibalization effects related to the corresponding elasticities.
- Sends anchor prices to RDF for reference purposes.
- Sends historical prices. RDF uses these to calculate the regular price lifts.

From RPO to RDF

Sends the price plan that allows RDF to calculate the three components of the regular price lift: regular price item effect, regular price cannibalization effect, and regular price halo effect.

From RDF to RPO

Sends forecasts to RPO. These forecasts represent the base demand at the item/store level. RPO aggregates the forecasts to the item/price zone level and uses that data to optimize prices.

ODI Integration

This chapter describes the integration of APC-RPO, RPO, and RDF using Oracle Data Integrator (ODI).

This chapter contains the following sections:

- [ODI Integration](#)

ODI Integration

Oracle Retail Enabled ODI Integration for APC-RPO, RPO, and RDF is a package that allows users to easily execute data transfers between these applications. It leverages ODI to store information about data interfaces among applications. ODI presents a user-friendly graphical interface for user-initiated data transfers and runtime monitoring. It also provides the ability to host application domains on different machines on a network, an ability not available with prior non-ODI integration strategies.

For more information about the ODI integration, see the *Oracle Retail Enabled Oracle Data Integrator Integration Implementation Guide*.

Measure Data Integration

The following data integration points for each application-to-application package are described in this section:

- [APC-RPO to RPO Package](#)
- [APC-RPO to RDF Package](#)
- [RDF to RPO Package](#)
- [RPO to RDF Package](#)

The scripts listed in each section are listed in the following directory:

`$RPAS_HOME/scripts/integration/ODI`

APC-RPO to RPO Package

The following information is about the APC-RPO to RPO integration package. This package sends anchor prices, self elasticities, and historical minimum and maximum prices from APC-RPO to RPO.

Data Mapping for APC-RPO to RPO Package

Data is sent when the expression is populated with a value other than the default value, which is zero.

Table 5–1 APC-RPO to RPO Data

| APC-RPO Expression | RPO Target Measure |
|--------------------|--------------------|
| exprtelt | ol1gamma |
| anchorprice | ol1anchprc |
| minhistprice | ol1hisloprc |
| maxhistprice | ol1hishiprc |

APC-RPO to RDF Package

The following information is about the APC-RPO to RDF integration package. This package sends anchor prices, self elasticities, and historical minimum and maximum prices from APC-RPO to RDF.

Data Mapping for APC-RPO to RDF Package

Data is sent when the expression is populated with a value other than the default value, which is zero.

Table 5–2 APC-RPO to RDF Data

| APC-RPO Expression | RDF Target Measure |
|--------------------|--------------------|
| achprstr | rdfanchnprc |
| exptelsstr | rdfgamma |
| price | rdfrprice |

RDF to RPO Package

The following information is about the RDF to RPO integration package. This package sends item-based demand from RDF to RPO.

Data Mapping for RDF to RPO Package

Data is sent when the expression is populated with a value other than the default value, which is zero.

Table 5–3 RDF to RPO Data

| RDF Expression | RPO Target Measure |
|----------------|--------------------|
| appf01xb | dl1itbdsp |

RPO to RDF Package

The following information is about the RPO to RDF integration package. This package sends item prices from RPO to RDF.

Data Mapping for RPO to RDF Package

Data is sent when the expression is populated with a value other than the default value, which is zero.

Table 5–4 RPO to RDF Data

| RPO Expression | RDF Target Measure |
|----------------|--------------------|
| fappitpc | rdprice |

Script Integration

This chapter describes the integration among APC-RPO, RPO, and RDF.

Integration Scripts

Integration scripts are used for moving data between applications. The following rules apply to integration scripts:

- The exportMeasure utility is used to export data in CSV (comma-separated values) format. This maintains the consistency of start and width attributes across different applications.
- Data exported from the source application is placed in the destination domain input directory.
- Export scripts must run before load scripts. They should be run in the batch window.
- Do not hard-code domain paths. The paths are entered as command-line arguments.

apcrpo_export.sh

The apcrpo_export.sh script exports measures from APC-RPO for consumption by RPO and RDF. Each measure is exported to

```
$APCRPO_MASTERDOMAIN/output/$Measure.csv.rpl.$TIMESTAMP
```

where,

- \$APCRPO_MASTERDOMAIN is given by the “-d” parameter
- \$Measure is taken from the first column in the Exported Measures table
- The \$TIMESTAMP is the current time in YYYYMMDD.HHMMSS format

The output files are in csv format.

[Table 6–1, "Parameters for apcrpo_export"](#) contains the details about the parameters for this script.

Table 6–1 Parameters for apcrpo_export

| Parameter Name | Required? | Input | Description |
|----------------|-----------|------------------|---|
| -d | Yes | File system path | The path to the top-level APC-RPO domain. |

Table 6–1 (Cont.) Parameters for `apcrpo_export`

| Parameter Name | Required? | Input | Description |
|----------------------------|-----------|------------------|--|
| <code>-maxprocesses</code> | No | Positive integer | A positive integer specifying how many parallel processes to use when generating the export. If neither this option nor <code>-noparallel</code> is specified, it defaults to the value in the environment variable <code>RPAS_PROCESSES</code> . If <code>RPAS_PROCESSES</code> is not set, it defaults to 3. |
| <code>-noparallel</code> | No | None | Instructs the script to run serially. Equivalent to “ <code>-maxprocesses 1</code> ” |

Table 6–2, “Exported Measures for `apcrpo_export`” contains details about the exported measures for this script.

Table 6–2 Exported Measures for `apcrpo_export`

| Measure Name | Intersection | Description |
|---------------------------|--------------------------|--|
| <code>exprteltst</code> | item/RHS item/price zone | Cross elasticities and self elasticities calculated by APC-RPO. This is sent to RPO. |
| <code>anchorprice</code> | item/price zone | The historical anchor price. This is sent to RPO. |
| <code>minhistprice</code> | item/price zone | The historical minimum price. This is sent to RPO. |
| <code>maxhistprice</code> | item/price zone | The historical maximum price. This is sent to RPO. |
| <code>maxprcuppct</code> | item/price zone | The maximum price increase percentage change. This is sent to RPO. |
| <code>maxprcdwnpct</code> | item/price zone | The maximum price decrease percentage change. This is sent to RPO. |
| <code>finstderr</code> | item/price zone | The self elasticity standard error. This is sent to RPO. |
| <code>fintstat</code> | item/price zone | The self elasticity T-Statistics. This is sent to RPO. |
| <code>exptelsstr</code> | item/RHS item/store | Cross elasticities and self elasticities calculated by APC-RPO. This is sent to RDF. |
| <code>achprstr</code> | item/store | The historical anchor price. This is sent to RDF. |
| <code>price</code> | week/item/store | The historical price. This is sent to RDF. |

Notes for `apcrpo_export.sh`

- Use the `apcrpo_export.sh` script to export the parameters that have been generated in APC-RPO to RPO. The script returns a error code of 0 when it completes successfully.
- Since the `apcrpo_export.sh` script exports values generated by `apcrpo_batch.sh`, you should generally run that script prior to exporting the APC-RPO parameters.
- You should set the following variables for the `apcrpo_export.sh` script:
 - `RPAS_PROCESSES` – this optional parameter takes an integer value that defaults to 3 if it is not set.
 - `APCRPO_PROCESSES` – this optional parameter takes an integer value that defaults to `$RPAS_PROCESSES` if it is not set.
 - `RPAS_HOME` – this is a mandatory parameter. You must also source `$RPAS_HOME/rpaslogin.ksh`.

Outputs for RPO

This section describes the following output files sent to RPO:

- [Cross Item Elasticity](#)
- [Anchor Price \(Optional\)](#)
- [Minimum History Price \(Optional\)](#)
- [Maximum History Price \(Optional\)](#)
- [Maximum Price Increase Percentage Change](#)
- [Maximum Price Decrease Percentage Change](#)
- [Self Elasticity Standard Error](#)
- [Self Elasticity T-Statistics](#)

Cross Item Elasticity

The cross item elasticity among the items. The location dimension depends on the optimization level the client will use. For instance, if a user sets the prices at item/price zone, then the location dimension is price zone. This measure includes item self price elasticity and halo and cannibalization elasticities inside.

Intersection: item/RHS item/location

[Table 6–3](#) describes the fields in this file.

Table 6–3 *Cross Item Elasticity Fields*

| Field | Description | Data Type | Importance |
|----------|---|-----------|------------|
| item | Retailer's Item ID | Text | Required |
| iter | RHS Item ID | Text | Required |
| location | The location dimension for the optimization level | Text | Required |
| value | Cross item elasticity value | Number | Required |

Example:

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

Anchor Price (Optional)

The anchor price is the baseline price used to calculate the price elasticity.

File Name: anchorprice.csv.rpl

Intersection: item/location

[Table 6–4](#) describes the fields in this file.

Table 6–4 Anchor Price Fields

| Field | Description | Data Type | Importance |
|----------|---|-----------|------------|
| item | Retailer's Item ID | Text | Required |
| location | The location dimension for the optimization level | Text | Required |
| value | Anchor price | Number | Required |

Example:

10033240, 1, 3.99

Minimum History Price (Optional)

The minimum historical regular price used in the price elasticity calculation. The minimum historical price is used in RPO for risk calculation.

File Name: minhistprice.csv.rpl

Intersection: item/location

[Table 6–5](#) describes the fields in this file.

Table 6–5 Minimum History Price Fields

| Field | Description | Data Type | Importance |
|----------|---|-----------|------------|
| item | Retailer's Item ID | Text | Required |
| location | The location dimension for the optimization level | Text | Required |
| value | Minimum history price | Number | Required |

Example:

10033240, 1, 5.69

Maximum History Price (Optional)

The maximum historical regular price used in the price elasticity calculation. The maximum historical price is used in RPO for risk calculation.

File Name: maxhistprice.csv.rpl

Intersection: item/location

[Table 6–6](#) describes the fields in this file.

Table 6–6 *Maximum History Price Fields*

| Field | Description | Data Type | Importance |
|----------|---|-----------|------------|
| item | Retailer's Item ID | Text | Required |
| location | The location dimension for the optimization level | Text | Required |
| value | Maximum history price | Number | Required |

Example:

10033240, 1, 8.99

Maximum Price Increase Percentage Change

The maximum price increase in the historical data.

File Name: maxprcuppcnt.csv.rpl

Intersection: item/price zone

[Table 6–7](#) describes the fields in this file.

Table 6–7 *Maximum Price Decrease Percentage Change Fields*

| Field | Description | Data Type | Importance |
|------------|---|-----------|------------|
| item | Retailer's Item ID | Text | Required |
| price zone | The price zone for the optimization level | Text | Required |
| value | Maximum history price | Number | Required |

Example:

10033240, 1, 0.105

Maximum Price Decrease Percentage Change

The maximum price decrease in the historical data.

File Name: maxprcdwnpct.csv.rpl

Intersection: item/price zone

[Table 6–8](#) describes the fields in this file.

Table 6–8 Maximum Price Decrease Percentage Change Fields

| Field | Description | Data Type | Importance |
|------------|---|-----------|------------|
| item | Retailer's Item ID | Text | Required |
| price zone | The price zone for the optimization level | Text | Required |
| value | Maximum history price | Number | Required |

Example:

10033240, 1, 0.05

Self Elasticity Standard Error

The standard error information when calculating the self price elasticity for the given item/price zone.

File Name: finstderr.csv.rpl

Intersection: item/price zone

[Table 6–9](#) describes the fields in this file.

Table 6–9 Self Elasticity Standard Error Fields

| Field | Description | Data Type | Importance |
|------------|---|-----------|------------|
| item | Retailer's Item ID | Text | Required |
| price zone | The price zone for the optimization level | Text | Required |
| value | Maximum history price | Number | Required |

Example:

10033240, 1, 1.69

Self Elasticity T-Statistics

The T-statistic information while calculating the self price elasticity for the given item/price zone.

File Name: fintstat.csv.rpl

Intersection: item/price zone

[Table 6–10](#) describes the fields in this file.

Table 6–10 Self Elasticity T-Statistics Fields

| Field | Description | Data Type | Importance |
|------------|---|-----------|------------|
| item | Retailer's Item ID | Text | Required |
| price zone | The price zone for the optimization level | Text | Required |
| value | Maximum history price | Number | Required |

Example:

10033240, 1, 1.96

Outputs for RDF

This section describes the following output files sent to RDF:

- [Cross Item and Self Elasticity](#)
- [Anchor Price](#)
- [Price](#)

Cross Item and Self Elasticity

The cross item elasticity and self elasticity among the items. The location dimension depends on the optimization level the user chooses. For instance, if a user sets the prices at item/price zone, then the location dimension is price zone. This measure includes item self price elasticity and halo and cannibalization elasticities inside.

Intersection: item/RHS item/location

[Table 6–11](#) describes the fields in this file.

Table 6–11 Cross Item and Self Elasticity Fields

| Field | Description | Data Type | Importance |
|----------|---|-----------|------------|
| item | Retailer's Item ID | Text | Required |
| iter | RHS Item ID | Text | Required |
| location | The location dimension for the optimization level | Text | Required |
| value | Cross item elasticity value | Number | Required |

Example:

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

Anchor Price

The anchor price is the baseline price used to calculate the price elasticity.

File Name:

Intersection: item/store

[Table 6–12](#) describes the fields in this file.

Table 6–12 Anchor Price Fields

| Field | Description | Data Type | Importance |
|----------|---|-----------|------------|
| item | Retailer's Item ID | Text | Required |
| location | The location dimension for the optimization level | Text | Required |
| value | Anchor price | Number | Required |

Example:

10033240, 1, 2.99

Price

The historical price for an item.

File Name: price.csv.rpl

Intersection: week/item/store

[Table 6–13](#) describes the fields in this file.

Table 6–13 Price Fields

| Field | Description | Data Type | Importance |
|-------|--------------------------------------|-----------|------------|
| item | Retailer's Item ID | Text | Required |
| week | The week for the optimization level | Text | Required |
| store | The store for the optimization level | Text | Required |
| value | Price value | Number | Required |

Example:

10033240,wk01,str101, 3.99

Configuration Considerations

This chapter provides information about changes that can be made to the APC-RPO configuration. If the default configuration of APC-RPO does not meet the needs of your business, you can configure it to match your organizational structure.

Hierarchies can only be used to determine the hierarchy aspects applicable directly to dimensions, attributes, facts, and escalation. Because of the RPAS limitations on intersection, distinct hierarchies must exist for the construction of all intersections in order to support all facts. No more than one dimension from any hierarchy can exist in a measure intersection.

This chapter contains the following sections:

- [Calendar \(CLND\) Hierarchy](#)
- [Product \(PROD\) Hierarchy](#)
- [Location \(LOC\) Hierarchy](#)
- [Escalation Levels \(ELH\) Hierarchy](#)

Calendar (CLND) Hierarchy

The Calendar hierarchy, shown in [Table 7-1](#), represents time in all RPAS solutions. It is a required hierarchy. RPAS requires a dimension named day (Day). This level is not displayed in the solution.

File name: clnd.csv.dat

File format: comma-separated values file

Table 7-1 *Calendar Hierarchy*

| Name | Label | Hierarchy Type | Child |
|----------|---------|----------------|-------|
| Year ID | Year | Main | Qtrtr |
| Qtrtr ID | Quarter | Main | Mnth |
| Mnth ID | Month | Main | Week |
| Week ID | Week | Main | Day |
| Day ID | Day | Main | None |

Product (PROD) Hierarchy

The Product Hierarchy, shown in [Table 7-2](#), represents the organizational levels of the retailer's merchandise.

File name: prod.csv.dat

File format: comma-separated values file

Table 7-2 Product Hierarchy

| Name | Label | Hierarchy Type | Child |
|-----------------|--------------|-------------------|--------------|
| Brand ID | Brand | Main or Alternate | Channel |
| Channel ID | Channel | Main or Alternate | Department |
| Department ID | Department | Main or Alternate | Category |
| Category ID | Category | Main or Alternate | Sub-category |
| Sub-category ID | Sub-category | Main or Alternate | Class |
| Class ID | Class | Main or Alternate | Item |
| Item ID | Item | Main or alternate | None |

Note: Any changes to this hierarchy must be accompanied by changes to all the elements that employ the particular level that is being modified or removed. Adding levels or branches or changing labels should not require any changes to the APC-RPO configuration.

Location (LOC) Hierarchy

The Location hierarchy, shown in [Table 7-3](#), represents the retailer's retail locations and their roll-ups.

File name: loc.csv.dat

File format: comma-separated values file

Table 7-3 Location Hierarchy

| Name | Label | Hierarchy Type | Child |
|---------------|------------|----------------|----------|
| Prize Zone ID | Price Zone | Alternate | Store |
| Company ID | Company | Main | Chain |
| Chain ID | Chain | Main | Channel |
| Channel ID | Channel | Main | Area |
| Area ID | Area | Main | Region |
| Region ID | Region | Main | District |
| District ID | District | Main | Store |
| Store ID | Store | Main | None |

Escalation Levels (ELH) Hierarchy

The Escalation Level Hierarchy, shown in [Table 7-4](#), represents aggregation levels for elasticity data. The elasticity information is generated at the base level and then aggregated to higher levels by APC-RPO.

File name: elh.csv.dat

File format: comma-separated values file

Table 7-4 Escalation Levels Hierarchy

| Name | Label | Hierarchy Type | Child |
|---------------------|------------------|----------------|-------|
| Escalation Level ID | Escalation Level | Main | None |

Time Periods Hierarchy

The Time Periods hierarchy, shown in [Table 7-5](#), represents a rough division of the historical data into two approximately equal-size sample populations. This information is used internally by the algorithm that calculates elasticity to generate and smooth the elasticities.

File name: tprd.csv.dat

File format: comma-separated values file

Table 7-5 Time Periods Hierarchy

| Name | Label | Hierarchy Type | Child |
|----------------|--------|----------------|-------|
| Time Period ID | Period | Main | None |

RHS Product Hierarchy

The RHS Product hierarchy, shown in [Table 7-6](#), is used by APC-RPO to calculate cross-item effects, including halo and cannibalization. It is the exact duplicate of the Product Hierarchy. Any changes to the Product Hierarchy must be duplicated here and in the corresponding load file.

File name: prod.csv.dat

File format: comma-separated values file

Table 7-6 RHS Product Hierarchy

| Name | Label | Hierarchy Type | Child |
|-----------------|--------------|-------------------|--------------|
| Brand ID | Brand | Main or Alternate | Channel |
| Channel ID | Channel | Main or Alternate | Department |
| Department ID | Department | Main or Alternate | Category |
| Category ID | Category | Main or Alternate | Sub-category |
| Sub-category ID | Sub-category | Main or Alternate | Class |
| Class ID | Class | Main or Alternate | Item |
| Item ID | Item | Main or alternate | None |

Batch Processing

This chapter contains a detailed description of the scripts used by APC-RPO. These scripts are used for the processes required to support APC-RPO, to generate the demand parameters, and load the generated measures.

This chapter contains the following sections:

- [apcrpo_batch.sh](#)
- [apcrpo_batch_local.sh](#)
- [buildAPCRPO.sh](#)
- [loadApcRpoMeasures.sh](#)

Scripts

At a high level, the batch process for generating the measures required for RPO and RDF consists of the following steps:

- Parameter default and override merging
- Historical data filtering
- Data transformation and filtering
- Calculation of raw self-elasticity
- Regularization of raw self-elasticity
- Calculation of halo-type cross elasticity
- Calculation of cannibalization-type cross elasticity
- Calculation of anchor and min/max historical prices

apcrpo_batch.sh

The `apcrpo_batch.sh` script runs the batch process to generate self elasticities, cross elasticities, halo effects, cannibalization, historical anchor, minimum prices, maximum prices, and the associated analysis data. Various parts of the process may be enabled or disabled in the Admin and Maintenance workbooks. The batch is run in all local domains.

Table 8–1, "Parameters for `apcrpo_batch.sh`" contains the details about the parameters for this script.

Table 8–1 Parameters for `apcrpo_batch.sh`

| Parameter Name | Required? | Input | Description |
|----------------------------|-----------|------------------|--|
| <code>-d</code> | Yes | File system path | The path to the top-level APC-RPO domain. |
| <code>-maxprocesses</code> | No | Positive integer | A positive integer specifying how many parallel processes to use when generating the export. If neither this option or <code>-noprogram</code> is specified, it defaults to the value in the environment variable <code>RPAS_PROCESSES</code> . If <code>RPAS_PROCESSES</code> is not set, it defaults to 3. |
| <code>-noprogram</code> | No | None | Instructs the script to run serially. Equivalent to " <code>-maxprocesses 1</code> " |

Notes for `apcrpo_batch.sh`

- Use the `apcrpo_batch.sh` script to generate the APC-RPO elasticities and other parameter information that will be used by RPO. The script returns a error code of 0 when it completes successfully.
- In order to use this script, you must have a working APC-RPO domain. The required historical data must have been loaded into the domain and the customer-specific batch parameters set.
- You must set the following variables for the `apcrpo_batch.sh` script:
 - `RPAS_PROCESSES` – this optional parameter takes an integer value that defaults to 3 if it is not set.
 - `APCRPO_PROCESSES` – this optional parameter takes an integer value that defaults to `$RPAS_PROCESSES` if it is not set.
 - `RPAS_HOME` – this is a mandatory parameter. You must also source `$RPAS_HOME/rpaslogin.ksh`.

apcrpo_batch_local.sh

The `apcrpo_batch_local.sh` script is identical to `apcrpo_batch.sh`, except that it only runs for one local domain. The purpose of the script is to eliminate the necessity to process all local domains.

Table 8–2, "Parameters for `apcrpo_batch_local.sh`" contains the details about the parameters for this script.

Table 8–2 Parameters for `apcrpo_batch_local.sh`

| Parameter Name | Required? | Input | Description |
|----------------|-----------|------------------|--|
| -d | Yes | File system path | The path to the sub-domain under the top-level APC-RPO domain. |

Notes for `apcrpo_batch_local.sh`

- Use the `apcrpo_batch_local.sh` script to generate the APC_RPO elasticities and other parameter information that will be used by RPO. Note that this script allows the information to be extracted for a single local domain at a time. The script returns a error code of 0 when it completes successfully.
- In order to use this script, you must have a working APC-RPO domain. The required historical data must have been loaded into the domain and the customer-specific batch parameters set.
- You must set the following variables for the `apcrpo_batch_local.sh` script:
 - `RPAS_HOME` – This is a mandatory parameter. You must also source `$RPAS_HOME/rpaslogin.ksh`.

buildAPCRPO.sh

The `buildAPCRPO.sh` script parameters provide the necessary locations and information needed to install APC-RPO.

The default installation mode is a full installation. In this case, a new APC-RPO domain is created from scratch.

The use of the `-p` flag causes the script to patch. In this case, changes are moved from the configuration into the domain.

The use of the `-t` parameter causes the script to build in test mode. In this case, the script checks that paths are correct, but does not perform the installation. This script calls the `loadApcRpoMeasures.sh` script.

Table 8–3, "Parameters for `buildAPCRPO.sh`" contains the details about the parameters for this script.

Table 8–3 Parameters for `buildAPCRPO.sh`

| Parameter Name | Required? | Input | Description |
|----------------|-----------|------------------|--|
| -c | No | File system path | The path to the configuration directory. Defaults to <code>\$PWD/../config</code> . |
| -d | No | File system path | The path to the top-level APC-RPO domain (existing or to be installed). Defaults to <code>\$PWD/../domain</code> . |
| -g | No | None | If present, use debug libraries when building/patching the domain. Otherwise, use optimized libraries. |

Table 8–3 (Cont.) Parameters for buildAPCRPO.sh

| Parameter Name | Required? | Input | Description |
|----------------|-----------|------------------|---|
| -i | No | File system path | The path to the input directory. Defaults to \$PWD/./input. |
| -l | No | File system path | The path to the log file. Defaults to \$PWD/./ |
| -m | No | Positive integer | Specify the maximum number of parallel processes to be used for the install. Defaults to 1. |
| -p | No | None | If present, make this a patch install instead of a full install. |
| -t | No | None | If present, make this a test install instead of a full install. |

Notes for buildAPCRPO.sh

- Use the buildAPCRPO.sh script to build or to apply a patch to an APC-RPO domain. The script returns a error code of 0 when it completes successfully. In addition, the log file (the “-l” parameter) will contain detailed information about the progress and status of the domain build.
- In order to use this script, you must have an APC-RPO configuration, a folder containing the historical measure data, and a directory in which to build the domain.
- You must set the following variables for the buildAPCRPO.sh script:
 - RPAS_HOME – this is a mandatory parameter. You must also source \$RPAS_HOME/rpaslogin.ksh.
 - RIDE_HOME – this is a mandatory parameter; it contains the path to the configuration tools. Note that \$RIDE_HOME/bin needs to be added to the PATH.

loadApcRpoMeasures.sh

The loadApcRpoMeasures.sh script loads measure data from the input file under the top-level domain directory.

Table 8–4, “Parameters for loadApcRpoMeasures” contains the details about the parameters for this script.

Table 8–4 Parameters for loadApcRpoMeasures

| Parameter Name | Required? | Input | Description |
|----------------|-----------|------------------|---|
| -d | No | File system path | The path to the top-level APC-RPO domain. Defaults to \$PWD/./domain/APCRPO. |
| -p | No | Positive integer | A positive integer specifying how many parallel processes to use when generating the export. Defaults to 1. |

Table 8–5, "Measures for loadApcRpoMeasures.sh" contains details about the exported measures for this script.

Table 8–5 Measures for loadApcRpoMeasures.sh

| Measure Name | Description |
|--------------|--|
| Maxprice | Historical maximum price data. |
| Minprice | Historical minimum price data. |
| Price | Historical price data. |
| Pos | Historical sales volume data. |
| Posdllr | Historical sales revenue data. |
| Promoind | Promotional indicator – when a particular item was promoted at a given location. |
| Pz2prlocmap | Mapping of item/stores to item/price zones. |

Notes for loadApcRpoMeasures.sh

- Use loadApcRpoMeasures script to load historical data into an APC-RPO domain. Although this script can be run standalone it is typically not. It is generally run by buildAPCRPO.sh.

When complete, this script writes the measure-loading status to the console. When this script is run as part of buildAPCRPO.sh, the output is redirected to a log file. The buildAPCRPO.sh script scans the resulting log file and reports exceptions or failures.

- You must set the following variables for the loadApcRpoMeasures.sh script:
 - RPAS_HOME – this is a mandatory parameter. You must also source \$RPAS_HOME/rpaslogin.ksh.
 - LOGLEVEL – this optional parameter defaults to warning.
 - RECORDLOGLEVEL – this optional parameter defaults to warning. The parameter controls the information generated at a per-record level.
 - APCRPO_LOCALIZATION_DIR – this optional parameter defaults to \$APCRPO_HOME/input_110n. If this parameter is set and corresponds to a directory that contains localization information, the script attempts to load localization data.
 - APCRPO_HOME – this mandatory parameter contains the path to the top level APCRPO domain.

Using the APC-RPO Plug-In

The APC-RPO plug-in, which automates part of the RPAS domain configuration, lets you to manage escalation levels. Using this data, the plug-in generates the appropriate intersections, measures, rule groups, rules, and workbooks. This chapter describes how to use the APC-RPO plug-in.

This chapter contains the following sections:

- [About the APCRPO Parameters Window](#)
- [Configuring the APC-RPO Solution](#)
- [Generating Intersections, Measures, Rules, and Workbooks](#)

In order to use the APC-RPO plug-in, you must configure the Escalation Levels Tab, which is accessed from the RPAS Configuration Tools.

About the APCRPO Parameters Window

From the RPAS Configuration Tools, select **Automation** from the menu bar, then **APCRPO**, and then **Specify Parameters**. The **APCRPO Parameters** window that contains the Escalation Levels Tab shown in [Figure 9–1, "Escalation Levels Tab"](#) is displayed.

Figure 9–1 Escalation Levels Tab

| Name | Label | Intersection | Path |
|------|---------------------|--------------|------|
| L01 | Escalation Level 01 | item_przn | 1 |
| L02 | Escalation Level 02 | class_przn | 2 |
| L03 | Escalation Level 03 | scat_przn | 3 |
| L04 | Escalation Level 04 | cat_przn | 4 |

The Escalation Levels tab is used to characterize each escalation level and its intersection.

Table 9–1, "Fields in the APC-RPO Escalation Levels Tab" describes the fields in the Escalation Levels Tab.

Table 9–1 Fields in the APC-RPO Escalation Levels Tab

| Name | Description |
|------------|---|
| Title Bar | The title of the window. |
| Tabs | The following tab is available: <ul style="list-style-type: none"> Escalation Levels Tab |
| Button Bar | The following buttons are available: <ul style="list-style-type: none"> Add – create a new default/blank escalation level in the grid area. Delete – delete the currently selected row in the grid area. |
| Grid Area | The area used to display the escalation levels and the intersections. The data is displayed in the following columns: <ul style="list-style-type: none"> Name – displays the name of each escalation level as it is generated. The names take the form of L01, L02, L03....L99. Label – provides a text entry box for the label for each escalation level. Intersection – provides access to the Select Intersection dialog box for each escalation level. Only product and location dimensions are allowed in the intersection. Path – provides a text entry box for the order in which the escalation values must be evaluated. The value must be a positive integer. |

Configuring the APC-RPO Solution

To configure the APC-RPO solution, from the RPAS Configuration Tools, select **Automation** from the menu bar, then **APCRPO**, and then **Specify Parameters**.

You must configure the following in the Escalation Levels Tab of the APCRPO dialog box: the Label, Intersection, and Path.

Configuring the Escalation Levels Tab

The Escalation Levels Tab is configured as follows:

1. To add an escalation level:
 - a. Click the **Add** icon in the Escalation Levels Tab. A new entry is added in the grid area.
 - b. Enter the label for the escalation level.
 - c. To set the escalation level intersection, click the intersection. The Select Intersection window is displayed.
 - d. Using the list options, set the intersection. Click **OK**.

For more information on setting intersections, see the *Oracle Retail Predictive Application Server Configuration Tools User Guide*.

2. To modify the intersection for an escalation level:
 - a. In the grid area, click the intersection of the escalation level to be modified. The Select Intersection window is displayed.
 - b. Using the list options, set the intersection. Click **OK**.
For more information on setting intersections, see the *Oracle Retail Predictive Application Server Configuration Tools User Guide*.
3. To delete an escalation level:
 - a. In the grid area, select the escalation level to be deleted from the configuration.
 - b. Click the **Delete** icon.

Generating Intersections, Measures, Rules, and Workbooks

The system automatically generates the intersections, measures, rules, and workbook templates for the retailer-specific configuration of APC-RPO. To start this process, click **OK** in the **APCRPO Parameters** window.

The system automatically generates the following:

- Intersections – All intersections necessary to support the base APC-RPO solution are created.
- Measures – All measures necessary to support the base APC-RPO solution are created.
- Rules – All rules and rule groups to support the base APC-RPO solution are created.
- Workbook Templates – All pre-defined workbook templates to support the base APC-RPO solution are created.

You can continue to make changes to the APC-RPO plug-in configuration, and the autogeneration process can be repeated as often as needed prior to the installation.

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. APC-RPO 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 the following:

- 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
- Demonstration data
- Training Materials

The user interface for APC-RPO has been translated from U.S. English into the following languages:

- Chinese (Traditional)
- Chinese (Simplified)
- Croatian
- Dutch
- French
- German
- Greek

- Hungarian
- Italian
- Japanese
- Korean
- Polish
- Portuguese (Brazilian)
- Russian
- Spanish (Spain)
- Swedish
- Turkish

Note: For information about adding languages for the first time or for translation information in general, see the *Oracle Retail Predictive Server Administration Guide for the Classic Client* or the *Oracle Retail Predictive Server Administration Guide for the Fusion Client*.

Index

A

anchor price, 6-4, 6-8
anchor price file, example, 6-4, 6-8
Anchorprice, 6-2
AnchPrc.csv.ovr, 6-4, 6-8
apcrpo_batch_local.sh, 8-3
apcrpo_batch.sh, 8-2
apcrpo_export.sh, 6-1

B

buildAPCRPO.sh, 2-4, 3-2, 3-11, 8-3

C

calendar hierarchy file, 3-3
calendar hierarchy file, example, 3-3
clnd.csv.dat, 3-3, 7-1
cross item elasticity, 6-3, 6-7
cross item elasticity file, example, 6-3, 6-7

E

elasticity calculation, 1-2, 8-1
elh.csv.dat, 3-5, 7-3
environment.sh, 3-2
escalation level hierarchy file, 3-5
escalation level hierarchy file, example, 3-5
escalation level intersection, 9-2
Escalation Levels tab, 9-1, 9-2
Exprteltst, 6-2

G

globaldomainconfig.xml, 3-2

H

HisHiPrc.csv.ovr, 6-5, 6-6
HisLoPrc.csv.ovr, 6-4, 6-8
historical data requirements, 2-1
historical data types, 2-1

L

loadApcRpoMeasures.sh, 8-3, 8-4

location hierarchy file, 3-4
location hierarchy file, example, 3-5
loc.csv.dat, 3-4, 7-2

M

mapping, price zone to item/store, 2-2
Maxhistprice, 6-2
maximum history price, 6-5, 6-7
maximum history price file, example, 6-5, 6-6, 6-7
maximum ticket price history, 3-9
maximum ticket price history file, example, 3-9
maxprice, 8-5
maxprice.csv.ovr, 3-9
merchandise hierarchy file, 3-4
merchandise hierarchy file, example, 3-4
Minhistprice, 6-2
minimum history price, 6-4
minimum history price file, example, 6-4, 6-8
minimum ticket price history, 3-9
minimum ticket price history file, example, 3-9
minprice, 8-5
minprice.csv.ovr, 3-9

O

out-of-stock indicator, 3-10
out-of-stock indicator file, example, 3-10
outstckindcsv.ovr, 3-10

P

pos, 8-5
pos.csv.ovr, 3-7
posdllr, 8-5
posdllr.csv.ovr, 3-7, 3-8
price, 8-5
price.csv.ovr, 3-8
prod.csv.dat, 3-4, 7-2
promoind, 8-5
promoindcsv.ovr, 3-10
promotion indicator history, 3-10
promotion indicator history file, example, 3-10
prop.csv.dat, 3-6
pz2prlocmap, 2-2, 8-5

R

rhs merchandise hierarchy file, 3-6
rhs merchandise hierarchy file, example, 3-6

S

sales dollars history, 3-7
sales dollars history file, example, 3-7
sales units history, 3-7
sales units history file, example, 3-7
store to price zone mapping, 3-8
store to price zone mapping file, example, 3-8

T

ticket price history, 3-8
ticket price history file, example, 3-8
time periods hierarchy file, 3-6
time periods hierarchy file, example, 3-6
tprd.csv.dat, 3-6, 7-3