

Oracle® Retail Item Planning

Operations Guide

Release 13.1.2.9

May 2010

Copyright © 2010, 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 Item Planning 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

Send Us Your Comments	xiii
Preface	xv
Audience.....	xv
Related Documentation.....	xvi
Customer Support	xvi
Review Patch Documentation	xvi
Oracle Retail Documentation on the Oracle Technology Network	xvii
Conventions	xvii
 1 Introduction	
Contents of this Guide	1-1
Overview of the Fashion Planning Bundle.....	1-1
Architecture of the Template Applications	1-2
Business Process Flow	1-3
Key Features of IP	1-3
Skills Needed for Implementation	1-4
Applications.....	1-4
Technical Concepts	1-4
 2 Implementation Considerations	
Historical Data	2-1
Sizing Impacts.....	2-1
Partitioning.....	2-2
Formatting.....	2-2
Plug-ins.....	2-2
Patch Considerations	2-3
Batch Scheduling.....	2-3
Security	2-3
Internationalization	2-4

3 Build Scripts

Installation Dependencies	3-1
Environment Setup	3-1
RPAS Installation	3-1
IP Installation	3-2
Custom Domain Build.....	3-2
Handling Common Hierarchy Files	3-2
Building the IP RPAS Domain	3-3
Batch Designs.....	3-4
Building a Domain.....	3-4
Patching a Domain.....	3-5
Creating Users and User Groups	3-5
Loading and Extracting Data	3-6

4 Data Flow

Overview of the IP Data Flow	4-1
Integration Interface Data Flow Description.....	4-2
From a Merchandising Application to Item Planning.....	4-2
From a Data Warehouse Application to Item Planning	4-2

5 Fashion Planning Bundle and ODI Integration

Overview of the Fashion Planning Bundle	5-1
Integration Interface Data Flow Description.....	5-2
From Size Profile Optimization to Assortment Planning	5-2
From Assortment Planning to Item Planning and an Allocation Application	5-2
From Item Planning to Merchandise Financial Planning	5-2
From Merchandise Financial Planning to Item Planning	5-2
From Merchandise Financial Planning to Assortment Planning.....	5-3
Execution Applications	5-4
Optimization Applications	5-4
Planning Applications.....	5-4
ODI Enabled Integration	5-5
Measure Data Integration	5-5
MFPCost to IP Package	5-6
Scripts Used By the MFPCost to IP Package.....	5-6
Data Mapping for MFPCost to IP Package	5-6
IP to MFPCost Package	5-7
Scripts Used By the IP to MFPCost Package.....	5-7
Data Mapping for IP to MFPCost Package	5-7
MFPRetail to IP Package	5-8
Scripts Used By the MFPRetail to IP Package.....	5-8
Data Mapping for MFPRetail to IP Package	5-8
IP to MFPRetail Package	5-9
Scripts Used By the IP to MFPRetail Package.....	5-9
Data Mapping for IP to MFPRetail Package	5-9
MFPCost to AP Package.....	5-10

Scripts Used By the MFPCost to AP Package	5-10
Data Mapping for MFPCost to AP Package.....	5-10
MFPRetail to AP Package	5-11
Scripts Used By the MFPRetail to AP Package.....	5-11
Data Mapping for MFPRetail to AP Package	5-11
SizeOpt to AP Package	5-11
Scripts Used By the SizeOpt to AP Package	5-11
Data Mapping for SizeOpt to AP Package	5-11
AP to IP Package	5-12
Scripts Used By the AP to IP Package.....	5-12
Data Mapping for AP to IP Package	5-12
Hierarchy Integration with ODI.....	5-13
Adding New Products: PROD Hierarchy Integration.....	5-13
Adding New Stores: LOC Hierarchy Integration.....	5-15

6 Script Integration

Integration Scripts.....	6-1
Batch Designs.....	6-2
Load Actuals Data.....	6-2
Export to MFP.....	6-3

7 Batch Processing

Batch Structure Overview.....	7-1
Batch Scheduling.....	7-1
Daily Batch Scripts	7-2
Weekly Batch Scripts	7-2
Unscheduled Administration Script.....	7-3
Batch Environment Scripts	7-3
Batch Designs	7-4
Formalization of DPM Positions	7-5
Calendar Hierarchy Load	7-6
Product Hierarchy Load.....	7-7
Location Hierarchy Load	7-8
Load On Order Data	7-9
Load Actuals Data.....	7-10
Load AP Data.....	7-12
Process Actuals Data.....	7-14
Run Batch Forecast.....	7-15
Refresh Workbooks.....	7-15
Auto Build Workbooks.....	7-16
Load Mapping Measures	7-17

8 Item Planning Configured for COE

Integration Methods and Communication Flow	8-1
Data Exported from COE to IP/RPAS	8-1
Forecast Summaries	8-1
Forecast Activities	8-2
Markdown Recommendations	8-4
Data Exported from IP/RPAS to COE	8-6
Business Rules	8-6
COE Initial EOL Salvage Value % Off	8-6
COE Initial EOL Exit Week	8-6
COE Initial EOL Sell Thru %	8-6
Example	8-6
Format of the Export File	8-6
Data Flow Between COE and IP/RPAS	8-8
Data Flow Description	8-8
RPAS Special Expression for COE	8-9
Overview on Evaluation of the Expression by RPAS	8-9
Environment Requirements	8-10
Configuration of the MDO_URL Measure	8-10
Security Features	8-10
Setup for Secure Calls to the Oracle WebLogic Server	8-10
Authentication	8-11
Confidentiality	8-12
Integrity	8-12
Setup for Secure Calls to Oracle Application Server	8-13
Authentication	8-13
Auditing	8-13
COE Expression Runtime Environment	8-14
Structure of the Special Expression	8-14
Input Arguments	8-15
Output Arguments	8-16
Batch Structure Overview	8-17
Transformation Scripts	8-17
Batch Scheduling	8-17
Daily Batch Scripts	8-18
Weekly Batch Scripts	8-18
Unscheduled Administration Script	8-19
Batch Environment Scripts	8-20
Batch Designs	8-20
Load COE Batch	8-20

List of Figures

1-1	Architecture of the RPAS Template Applications.....	1-2
1-2	Business Process Workflow	1-3
4-1	Data Flow for the Integration of IP with Other Applications	4-1
5-1	Conceptual Overview.....	5-1
5-2	Overview of Fashion Planning Bundle Integration	5-3
8-1	Data Flow Between COE and IP/RPAS	8-8

List of Tables

3-1	Build Scripts.....	3-3
3-2	Building a Domain Error Information	3-4
3-3	Patching a Domain Error Information	3-5
5-1	MFP Cost to IP: CP Approved Data.....	5-6
5-2	MFP Cost to IP: OP Approved Data.....	5-6
5-3	IP to MFPCost: CP Approved Data.....	5-7
5-4	IP to MFPCost: OP Approved Data.....	5-7
5-5	MFPRetail to IP: CP Approved Data.....	5-8
5-6	MFPRetail to IP: OP Approved Data	5-8
5-7	IP to MFPRetail: CP Approved Data.....	5-9
5-8	IP to MFPRetail: OP Approved Data	5-10
5-9	MFPCost to AP Data.....	5-10
5-10	MFPRetail to AP Data	5-11
5-11	SizeOpt to AP Data	5-11
5-12	AP to IP Data	5-12
5-13	Prod.csv File Dimension Positions/Labels	5-13
5-14	Loc.csv File Dimension Positions/Labels	5-15
6-1	Integration Scripts.....	6-2
6-2	Load Actuals Data Usage.....	6-2
6-3	Load Actuals Data Error Information.....	6-2
6-4	Export to MFP Error Information	6-3
7-1	Directories Used by Batch Scripts.....	7-1
7-2	Daily Batch Script.....	7-2
7-3	Weekly Batch Scripts	7-2
7-4	Unscheduled Administration Script	7-3
7-5	Formalization of DPM Positions Error Information	7-5
7-6	Calendar Hierarchy Load Error Information.....	7-6
7-7	Product Hierarchy Load Error Information.....	7-7
7-8	Location Hierarchy Load Error Information	7-8
7-9	Load Actuals Data Error Information.....	7-9
7-10	Load Actuals Data Error Information.....	7-11
7-11	Load Actuals Data Error Information.....	7-13
7-12	Process Actuals Error Information	7-14
7-13	Run Batch Forecast Error Information.....	7-15
7-14	Refresh Workbooks Error Information.....	7-15
7-15	Auto Build Workbooks Error Information.....	7-16
7-16	Load Actuals Data Error Information.....	7-17
8-1	Layout of the Forecast Summaries File.....	8-1
8-2	Forecast Summaries File After Transformation	8-2
8-3	Layout of the Forecast Activities File.....	8-2
8-4	Forecast Activities File After Transformation.....	8-3
8-5	Layout of the Markdown Recommendations File.....	8-4
8-6	Markdown Recommendations File After Transformation	8-5
8-7	COE Load File Description.....	8-7
8-8	COE Special Expression Input Arguments	8-15
8-9	COE Special Expression Output Arguments	8-16
8-10	Directories Used by Batch Scripts.....	8-17
8-11	Daily Batch Scripts.....	8-18
8-12	Weekly Batch Scripts	8-18
8-13	Unscheduled Administration Script	8-19
8-14	Load Actuals Data Error Information.....	8-21

Send Us Your Comments

Oracle Retail Item Planning Operations Guide, Release 13.1.2.9

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

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

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

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

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

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

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

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

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

Preface

Oracle Retail Operations Guides are designed so that you can view and understand the application's 'behind-the-scenes' processing, including such information as the following:

- Key system administration configuration settings
- Technical architecture
- Functional integration dataflow across the enterprise
- Batch processing

Audience

Anyone who has an interest in better understanding the inner workings of Item Planning can find valuable information in this guide. There are three audiences in general for whom this guide is written:

- System analysts and system operation personnel:
 - who are looking for information about Item Planning processes internally or in relation to the systems across the enterprise.
 - who operate Item Planning on a regular basis.
- Integrators and implementation staff who have the overall responsibility for implementing Item Planning into their enterprise.
- Business analysts who are looking for information about processes and interfaces to validate the support for business scenarios within Item Planning and other systems across the enterprise.

Related Documentation

For more information, see the following documents in the Oracle Retail Item Planning Release 13.1.2.9 documentation set:

- *Oracle Retail Item Planning Release Notes*
- *Oracle Retail Item Planning Installation Guide*

For more information about the Fashion Planning Bundle applications see the following documentation sets:

- Oracle Retail Item Planning Configured for COE documentation
- Oracle Retail Clearance Optimization Engine documentation
- Oracle Retail Assortment Planning documentation
- Oracle Retail Merchandise Financial Planning documentation
- Oracle Retail Size Profile Optimization documentation

For more information about RPAS and the ODI Enabled Integration, see the following documentation sets:

- Oracle Retail ODI Enabled Integration documentation
- Oracle Data Integrator documentation
- Oracle Retail Predictive Application Server documentation

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, 13.1) or a later patch release (for example, 13.1.2). 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

Oracle Retail Item Planning (IP) facilitates the translation of merchandise financial plans into execution level, item driven strategies to be followed throughout the life of a product. It represents the bottom-up planning process, complementing and working in concert with the top-down financial plans. By providing weekly visibility into item performance and financial targets, the item planning process increases the likelihood that merchandising strategies are successfully executed within the financial plan parameters. For a more detailed overview of the functionality within IP, see the *Oracle Retail Item Planning User Guide*.

Contents of this Guide

This operations guide addresses the following topics:

- [Chapter 1, "Introduction"](#). Overview of the IP 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 Scripts"](#). Information on building and patching the IP RPAS domain.
- [Chapter 4, "Data Flow"](#). Overview of the IP data flow.
- [Chapter 5, "Fashion Planning Bundle and ODI Integration"](#). Overview of the Fashion Planning Bundle data flow and the ODI enabled integration for the bundle.
- [Chapter 6, "Script Integration"](#). Overview of the script integration used for the Fashion Planning Bundle.
- [Chapter 7, "Batch Processing"](#). Explanation of batch scheduling and batch designs.
- [Chapter 8, "Item Planning Configured for COE"](#). Information on the integration and batch processing.

Overview of the Fashion Planning Bundle

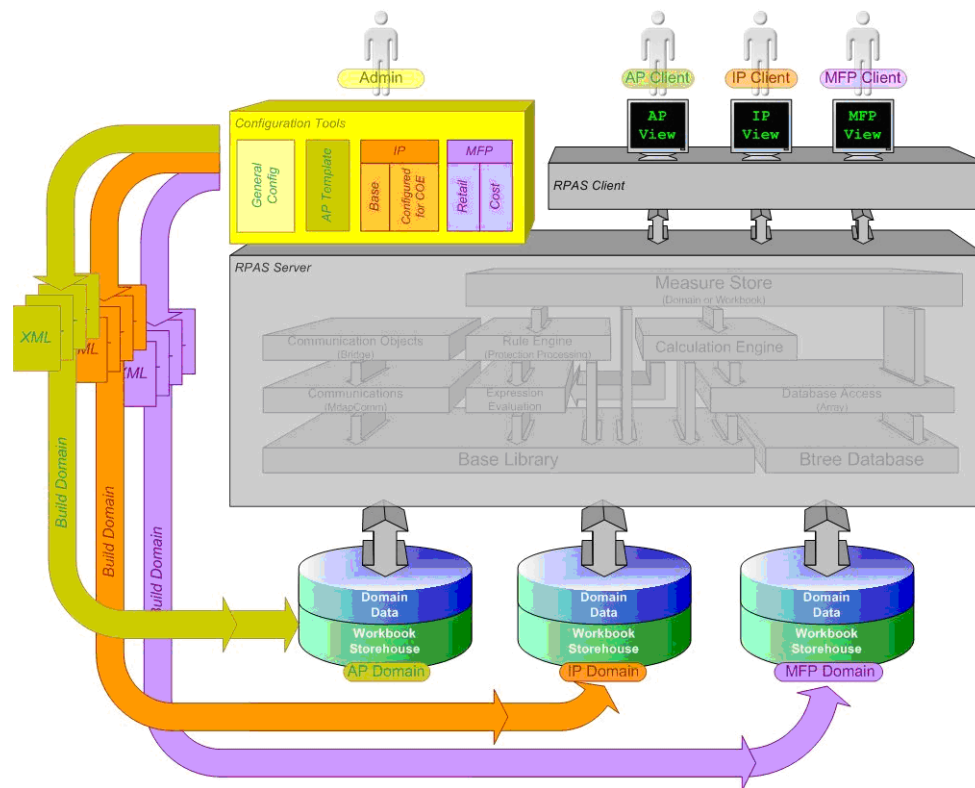
The Fashion Planning Bundle is the integration of Item Planning (IP), Clearance Optimization Engine (COE), Assortment Planning (AP), Merchandise Financial Planning (MFP), and Size Profile Optimization (SPO) as a full-suite planning solution for fashion retailers.

For details on the integration, see [Chapter 5](#) and [Chapter 6](#).

Architecture of the Template Applications

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

Figure 1–1 Architecture of the RPAS Template Applications



This diagram describes the RPAS template applications. In the truest sense, these 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 templates are pre-defined means by which 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 pre-defined attributes:

- Measures
- Special expressions
- Rules
- Workbook layout

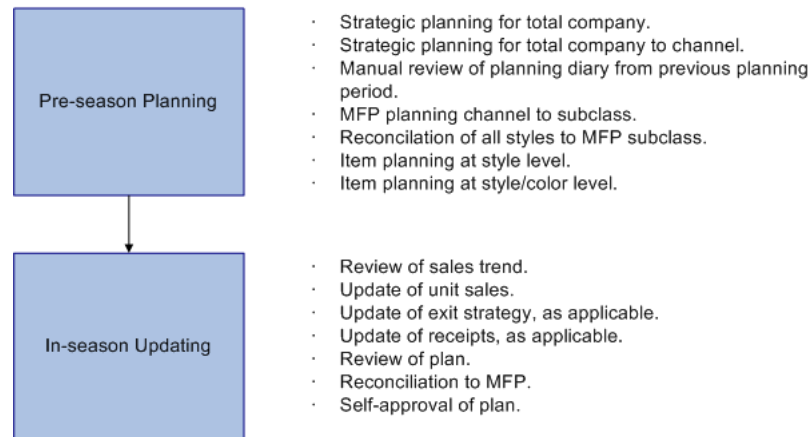
A client requests to use one of the templates via the Configuration Tools. A number of XML files are then output which define the domain to be created. These XML files are used to build the specified domain incorporating all the attributes mentioned above that have been defined specifically.

Once the domain has been created, the end user is able to access the domain data via the RPAS client. Based on the template used, workbooks, measures, rules, and so on are available to the end user.

Business Process Flow

Figure 1–2 shows a typical workflow for IP.

Figure 1–2 Business Process Workflow



Key Features of IP

IP provides the following features:

- Strategic planning for the total company
- Item planning at style and style/color level
- Review of sales trends and plans
- Update of unit sales, exit strategies, and receipts
- Reconciliation to MFP

Skills Needed for Implementation

The implementer needs an understanding of the following applications and technical concepts.

Applications

The implementer should understand interface requirements of the integrated applications and data sources for the master data, demand, and inventory history. For the Fashion Planning Bundle, the implementer needs this knowledge for the following applications:

- Oracle Retail Predictive Application Server
- Oracle Retail Assortment Planning
- Oracle Retail Item Planning
- Oracle Retail Clearance Optimization Engine
- Oracle Retail Merchandise Financial Planning
- Oracle Retail Size Profile Optimization
- Oracle Retail Data Warehouse (RDW)
- Oracle Retail Merchandising System (RMS)

Technical Concepts

The implementer should understand the following technical concepts:

- UNIX system administration, shell scripts, and job scheduling.
- Performance constraints based on the retailer's infrastructure
- Technical architecture for IP
- Retailer's hierarchical (SKU/store/day) data
- IP batch processes
- 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

Implementation Considerations

The following information needs to be considered before configuring IP:

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

Historical Data

It is recommended that you have at least two years of historical sales and inventory data for creating item plans. Less data can be used, but the more data that is available, the more statistical significance can be given to the item plans.

It is also important to have at least two years of history to support the embedded forecast functionality within IP. With less than two years of history, the forecast may not generate the optimal results.

Sizing Impacts

The following factors can affect size requirements:

- **Style/color**—number of style/colors. A style/color is the combination of the style information, such as fleece jacket, and color, such as navy.
- **Store**—number of physical, Web, and other distinct retail outlets. Although IP does not go down to the store level, data may be loaded at this level.
- **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.

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 IP domain is defined as a global domain. For performance reasons, a single 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, this will 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?

Note: The partitioning level in the IP configuration is Group. It is recommended that this not be changed.

Formatting

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

- In the configuration, 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*.
- To set up worksheet sizes and placements, exception value formatting, gridlines, and other formatting, once the domain is built, the implementer or a user can instantiate a workbook of the template to set up specific formatting and use the Format menu. The implementer can then save the format 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 IP.

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 IP 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 straightforward 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.

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. Batch scripts are used for importing and exporting data and for generating item plans. The retailer needs to decide the best time for running batch scripts within the available batch window.

How often to upload updated sales and inventory data and how often to recreate item plans needs to be determined.

- You have to consider at what interval to load the latest sales and inventory data. A weekly load of transactional type data is supported since the base intersection is at week. It is recommended that the information coming from other Fashion Bundle application be loaded daily.
- Product availability and seasonal changes can be reasons for recalculating the item plans. It can also be triggered by the addition of new products and availability of substantial new sales and inventory history.

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 is not 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—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

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, among others:

- 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 files
- Reports
- Demo data
- Training materials

The RPAS platform supports associated solution extensions and solution templates.

- A solution extension includes a collection of code and generally available configurations. Typically, solution extensions are implemented by a retailer with minimal configuration.
- A solution template does not include code. A solution template is most typically implemented as a retailer configuration.

Oracle Retail releases the translations of the RPAS server and client, as well as strings from the solution extensions.

Translations of the solution templates, including IP, are not released. Since the labels for measures, hierarchies, dimensions, and workbook templates, are typically changed by the retailer at implementation time, languages for the templates are not released. All templates have the ability to support multibyte characters.

Build Scripts

This chapter describes the scripts that must be executed to build or patch the IP RPAS domain.

Installation Dependencies

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

- For information on installing RPAS, see the *Oracle Retail Predictive Application Server Installation Guide*.
- For information on installing IP, see the *Oracle Retail Item Planning 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 ITEM_HOME. The UNIX user performing the installation needs to set up an environmental variable called ITEM_HOME in the user's profile:

```
export ITEM_HOME=<full path name to IP home>
```

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

IP Installation

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

The IP installer performs the following functions:

- Downloads the configuration and batch scripts into the `$ITEM_HOME/config` and `$ITEM_HOME/bin` directories
- Downloads a set of sample hierarchy and data files into the `$ITEM_HOME/input` directory
- Builds a sample domain at `$ITEM_HOME/domain/itemplan`

Custom Domain Build

To do a custom build of a domain:

1. Change to the configuration directory: `cd $ITEM_HOME/config`
2. Update the `globaldomainconfig.xml` file with the correct domain paths. Update the partition position information to correspond to the product hierarchy used in the file.
3. Change to the bin directory: `cd $ITEM_HOME/bin`
4. Update the `localdomainlist.cfg` file with the correct paths for the local domains.
5. If needed, update the default environment variables in `environment.ksh`.
6. Execute the `build_item.ksh` script: `./build_item.ksh`
7. It is expected that the first time `build_item.ksh` is executed, an error occurs when it tries to remove the old log file because a log file does not yet exist.

Handling Common Hierarchy Files

The following hierarchy files contain the superset of all the dimensions along the product, location, and calendar hierarchies:

- `prod.hdr.csv.dat`
- `loc.hdr.csv.dat`
- `clnd.hdr.csv.dat`

These common hierarchy files are shared among the Fashion Planning Bundle applications: AP, IP, MFP, and SPO.

Note: The common hierarchy files are not available for IP Configured for COE.

Each `hdr.csv.dat` (hdr) hierarchy file contains a header line that lists all the dimensions for which position information is contained in the file. The RPAS build process handles these `hdr` files, so that every application extracts the position information relevant to itself and ignores dimensions not configured in the application.

The `filterHier` utility is run on the `hdr` files to convert them into standard hierarchy files that are then passed to `loadHier`. The build process, which uses `rpasInstall`, can differentiate between standard and `hdr` hierarchy files. There is no need for the implementer to make any changes in the domain build process.

If using hdr files, the implementer needs to run filterHier before running loadHier. The filterHier utility converts the hdr files into standard hierarchy files that can be processed by loadHier. Note that there is no need to run filterHier if the standard hierarchy files are already available.

Note: The hdr files must reside outside the domain input directory before running filterHier. By default, the filterHier utility puts the newly created filtered hierarchy files into the input folder of the domain.

See the *Oracle Retail Predictive Application Server Administration Guide* for details on the RPAS utilities.

Building the IP RPAS Domain

The following scripts are not part of a normal batch schedule but are executed only to build or patch a domain.

[Table 3–1](#) outlines the build scripts. These scripts are located in the \$ITEM_HOME/bin directory.

Table 3–1 Build Scripts

Description	Script Name	Dependency
Build a domain	build_item.ksh	None
Patch a domain when the configuration changes do not affect formatting	patch_item_keepformats.ksh	None
Patch a domain when the configuration changes require updates to formatting	patch_item_deleteformats.ksh	None

Batch Designs

This section contains detailed information on the following build scripts:

- [Building a Domain](#)
- [Patching a Domain](#)

Building a Domain

Script

build_item.ksh

Usage

build_item.ksh

Error Information

Table 3–2 Building a Domain Error Information

Task Name	Error Code	Abort Required?	Description of Error
build_item	-1	yes	Error during domain build.
build_item	-2	yes	Error when adding default users.
build_item	-3	yes	Error during initial measure load.
build_item	-4	yes	Error found when running the batch calculations in the master domain.
build_item	-5	yes	The localdomainlist.cfg file was not found.
build_item	-6	yes	Error when running the batch calculations over the local domains.
build_item	-7	yes	Error during the disabling of commit later.
build_item	-8	yes	Error when updating the local domain paths to relative paths.

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 following RPAS utilities: usermgr, mace, loadmeasure, copyDomain, and domainprop. See the *Oracle Retail Predictive Application Server Administration Guide* for details on these utilities.
- All hierarchy and measure files are placed into the \$ITEM_HOME/input directory.
- The commit later functionality and insert measure functionality are disabled.

Patching a Domain

When changes have been made to IP that require changes to the domain, the following scripts are used to apply the patch to the domain.

Script

patch_item_keepformats.ksh
or
patch_item_deleteformats.ksh

Usage

patch_item_keepformats.ksh
patch_item_deleteformats.ksh

Error Information

Table 3–3 Patching a Domain Error Information

Task Name	Error Code	Abort Required?	Description of Error
patch_item_keepformats patch_item_deleteformats	-10	yes	Could not find the localdomainlist.cfg file.
patch_item_keepformats patch_item_deleteformats	-11	yes	Error during domain patch.

Notes

- Prior to RPAS 13.1.1, formats were not transferrable across major versions, and users had to delete formats before patching and then reapply the formats after the patch was installed. RPAS 13.1.1 included an upgrade to the save format process that allows RPAS applications to keep the formats when updating versions. Because this update is not backwards compatible, the script you use depends on the version of RPAS you are upgrading from:
 - If you are upgrading from an RPAS version prior to 13.1.1, use patch_ap_deleteformats. This script removes all existed saved formats (template, group, and user). Once you have installed the patch, reapply your formats.
 - If you are upgrading from RPAS 13.1.1 or later, use patch_ap_keepformats. The formats are automatically converted.
- The scripts use the Configuration Tools rpasInstall utility to build a domain. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The scripts use the Configuration Tools rpasInstall utility to build a domain. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.

Creating Users and User Groups

For greater security, users and user groups are not automatically created when you build or patch a domain. To create users and user groups, you must use the usermgr utility. To learn more about usermgr, see the Operational Utilities chapter of the *Oracle Retail Predictive Application Server Administration Guide*.

Loading and Extracting Data

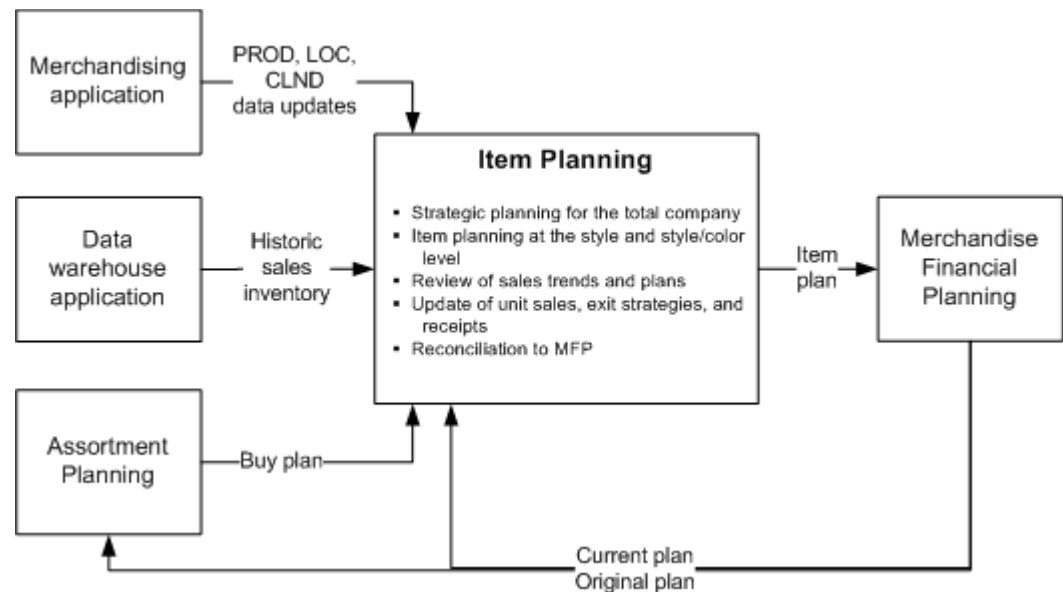
Data is loaded into IP using the standard RPAS approach. See the *Oracle Retail Predictive Application Server Administration Guide* for details on formatting the load data files and on the utilities that enable administrators to load data into RPAS. If you are using the ODI enabled integration, see [Chapter 5](#). If you are using script integration for the bundle, see [Chapter 6](#). For information on any other batch script, see [Chapter 7](#).

This chapter describes the data flow of Item Planning with other applications.

Overview of the IP Data Flow

Figure 4–1 shows the integration of IP with other applications and the flow of data between the applications.

Figure 4–1 Data Flow for the Integration of IP with Other Applications



Integration Interface Data Flow Description

These descriptions explain some of the data flows shown in [Figure 4–1](#). For information on the integration among IP, AP, and MFP, see [Chapter 5](#). For information on the interface between IP/RPAS and COE, see [Chapter 8](#).

From a Merchandising Application to Item Planning

Data for the following hierarchies is imported into IP from a merchandising application:

- Product (PROD) hierarchy
- Location (LOC) hierarchy
- Calendar (CLND) hierarchy

From a Data Warehouse Application to Item Planning

The following data is imported into IP from a data warehouse application:

- Historic sales
- Inventory

Fashion Planning Bundle and ODI Integration

This chapter describes the overall flow of data among the Fashion Planning Bundle applications, the integration among the Fashion Planning Bundle applications using Oracle Data Integrator (ODI), and the integration of hierarchies using ODI.

For information about script integration for the Fashion Planning Bundle, see [Chapter 6](#).

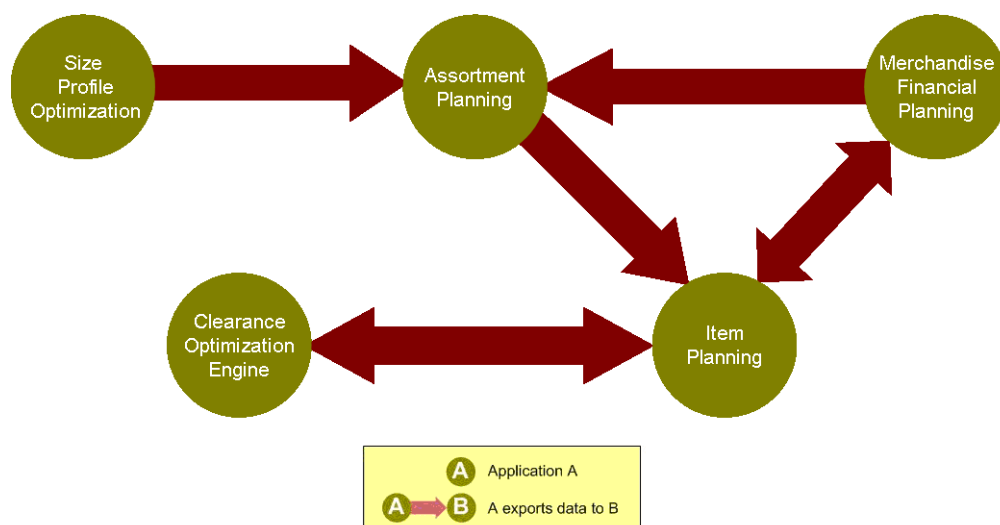
Overview of the Fashion Planning Bundle

This section describes the integration between solutions within the Fashion Planning Bundle. It does not describe external integrations.

The Fashion Planning Bundle is a full-suite planning solution for fashion retailers that integrates the following: Item Planning (IP), Clearance Optimization Engine (COE), Assortment Planning (AP), Merchandise Financial Planning (MFP), and Size Profile Optimization (SPO).

[Figure 5–1](#) shows the conceptual overview of the integration of these products.

Figure 5–1 Conceptual Overview



This solution supports data sharing among these applications. Note that the data sharing functionality is not dependent on the presence of all these applications. The defined data sharing between any of the applications works for the entire suite as well as for a subset of the applications.

Integration Interface Data Flow Description

These descriptions explain each of the data flows in [Figure 5–1](#).

From Size Profile Optimization to Assortment Planning

The following data is imported from SPO to AP:

- Store-level size profiles (current and archived versions) with corresponding escalation levels
- Store-clustered size profiles
- Pre-pack details: labels, descriptions, and configurations

The size profiles are by AP to obtain more detailed buy plans at style-color-size or pre-pack per store per receipt week. This enables the product to be tailored to the consumer selling patterns at each specific location.

SPO can group stores based on historical data into clusters, and these clusters can be used as a basis for creating Pre-packs.

For more information on SPO, see the Oracle Retail Size Profile Optimization documentation.

From Assortment Planning to Item Planning and an Allocation Application

The following data is exported from AP to IP and an allocation application:

- Buy plan

AP sends the buy plan to IP and the allocation application. The allocation application can use the sales plan or receipt plan created by AP to determine what inventory is allocated to stores. IP uses the pre-buying period assortment plan to track the performance of items during the buying period and to provide a framework to respond accordingly.

From Item Planning to Merchandise Financial Planning

The following data is exported from IP to MFP:

- Item plan

Approved data is exported from IP and loaded into MFP at the subclass level. The aggregation takes place within IP when exporting. Only the plans that have been approved since the last export in IP are imported into MFP.

MFP users can review and reconcile their merchandise financial plans (which are planned at the subclass level) to the approved item plans from IP.

From Merchandise Financial Planning to Item Planning

The following data is exported from MFP to IP:

- MFP current and original plans

The current and original plans are used by IP to help with the creation of item plans.

From Merchandise Financial Planning to Assortment Planning

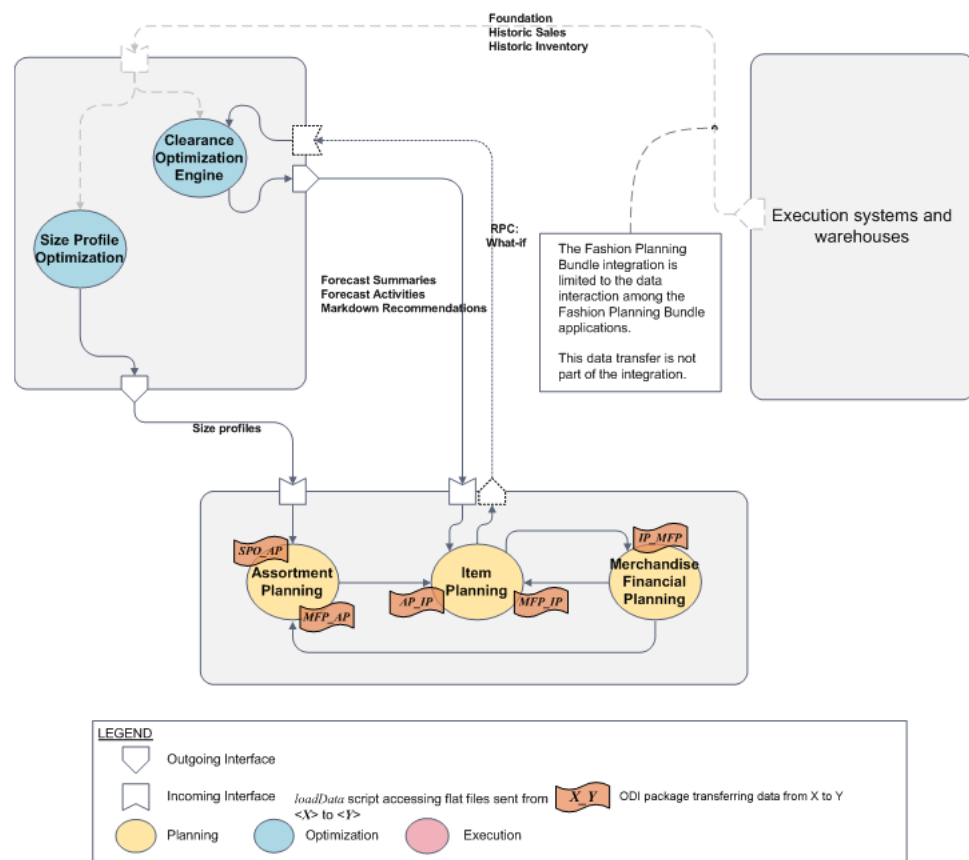
The following data is exported from MFP to AP:

- MFP current plans

MFP current plans are critical inputs into the AP process. They define the financial goals for that buying period. AP uses these plans as targets to determine the number of options it should carry, the number of weeks it should carry them, and the amount of markdown it can afford in order to meet the targets for sales and gross margin.

Figure 5–2 shows the applications and data flow that are part of the Fashion Planning Bundle.

Figure 5–2 Overview of Fashion Planning Bundle Integration



The applications shown in the diagram are in the following categories:

- Execution Applications
- Optimization Applications
- Planning Applications

Execution Applications

Note: Full data transfers from the execution applications are not part of the Fashion Planning Bundle integration. They are included in the diagram to illustrate that historical data needs to be obtained from applications outside the applications included in the Fashion Planning Bundle.

The execution applications provide the foundation and historical data that is input to the optimization applications.

- A data warehouse application provides the initial load of sales and inventory data.
- A merchandising application provides the daily and weekly data updates, or deltas, of the sales and inventory data. It provides data updates of master information such as styles, prices, stores, and so on.

Optimization Applications

The optimization applications take the foundation and historical data and produce size and markdown recommendations that are used by the planning applications.

- SPO creates profiles of the optimal size distribution by both merchandise category and store. This gives insight into consumer demand patterns by size. These size profiles feed into AP.

For more information on SPO, see the Oracle Retail Size Profile Optimization documentation. For more information on AP, see the Oracle Retail Assortment Planning documentation.

- COE provides markdown recommendations and forecasts that enable retailers to make informed markdown decisions. What-if data feeds back into COE from IP. COE is only available with IP configured for COE.

For more information on COE, see the Oracle Retail Clearance Optimization Engine documentation. For more information on IP, see the Oracle Retail Item Planning documentation.

Planning Applications

The planning applications take data from the optimization applications and create plans for the retailer.

- AP creates buy plans for the retailer based on size profiles from SPO and the subclass plans from MFP. The buy plans feeds into IP.
- IP takes the buy plan from AP and the subclass plan from MFP to create an item plan. When configured for COE, markdown data from COE is also used. IP output feeds back into MFP. What-if data feeds back into COE to create new markdown plans.
- MFP takes data from IP which is consulted when creating the subclass plan in MFP. Based on financial goals, the subclass plans determine how much merchandise should be available in a store in order to meet the goals. Subclass plans are fed into AP and IP.

ODI Enabled Integration

Oracle Retail ODI Enabled Integration for the Fashion Planning Bundle is a set of packages that allows users to easily execute data transfers between Fashion Planning Bundle applications.

Oracle Retail ODI Enabled Integration leverages Oracle Data Integrator (ODI) to store information about data interfaces among Fashion Planning Bundle 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.

The Fashion Planning Bundle applications supported by the ODI enabled integration include Merchandise Financial Planning Retail (MFP Retail), Merchandise Financial Planning Cost (MFP Cost), Item Planning (IP), Item Planning Configured for COE (IP COE), Assortment Planning (AP), and Size Profile Optimization (SPO).

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

Measure Data Integration

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

- [MFPCost to IP Package](#)
- [IP to MFPCost Package](#)
- [MFPRetail to IP Package](#)
- [IP to MFPRetail Package](#)
- [MFPCost to AP Package](#)
- [MFPRetail to AP Package](#)
- [SizeOpt to AP Package](#)
- [AP to IP Package](#)

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

`$RPAS_HOME/scripts/integration/ODI`

MFP Cost to IP Package

The following information is about the MFP Cost to IP integration.

Scripts Used By the MFP Cost to IP Package

- mfp_to_ip_init.ksh
- mfp_to_ip_finalize.ksh

Data Mapping for MFP Cost to IP Package

The CP Approved data is sent when mowpappcpnewb is set to TRUE.

Table 5–1 MFP Cost to IP: CP Approved Data

MFP Cost Formula	IP Target Measure
bucpbopc	bucpbopc
bucpbopu	bucpbopu
bucpeopc	bucpeopc
bucpeopu	bucpeopu
bucprecc	bucprecc
bucprecu	bucprecu
bucpgmpv	bucpgmpv
bucpslsc	bucpslsc
bucpslsr	bucpslsr
bucpslsu	bucpslsu

The OP Approved data is sent when mowpappopnewb is set to TRUE.

Table 5–2 MFP Cost to IP: OP Approved Data

MFP Cost Formula	IP Target Measure
buopbopc	buopbopc
buopbopu	buopbopu
buopeopc	buopeopc
buopeopu	buopeopu
buoprecc	buoprecc
buoprecu	buoprecu
buopgmpv	buopgmpv
buopslsc	buopslsc
buopslsr	buopslsr
buopslsu	buopslsu

IP to MFPCost Package

The following information is about the IP to MFPCost integration.

Scripts Used By the IP to MFPCost Package

- ip_to_mfp_init.ksh
- ip_to_mfp_finalize.ksh

Data Mapping for IP to MFPCost Package

The CP Approved data is sent when ipcpappnewb is set to TRUE.

Table 5–3 IP to MFPCost: CP Approved Data

IP Formula	MFP Cost Target Measure
ipcpbopc	ipcpbopc
ipcpbopu	ipcpbopu
ipcpeopc	ipcpeopc
ipcpeopu	ipcpeopu
ipcpgmpv	ipcpgmpv
ipcprecc	ipcprecc
ipcprecu	ipcprecu
ipcpslsr	ipcpslsr
ipcpslsu	ipcpslsu
ipcpslsc	ipcpslsc

The OP Approved data is sent when ipopappnewb is set to TRUE.

Table 5–4 IP to MFPCost: OP Approved Data

IP Formula	Target MFP Cost Measure
ipopbopc	ipopbopc
ipopbopu	ipopbopu
ipopeopc	ipopeopc
ipopeopu	ipopeopu
ipopgmpv	ipopgmpv
ipoprecc	ipoprecc
ipoprecu	ipoprecu
ipopslsr	ipopslsr
ipopslsu	ipopslsu
ipopslsc	ipopslsc

MFPRetail to IP Package

The following information is about the MFPRetail to IP integration.

Scripts Used By the MFPRetail to IP Package

- mfp_to_ip_init.ksh
- mfp_to_ip_finalize.ksh

Data Mapping for MFPRetail to IP Package

The CP Approved data is sent when mowpappcpnewb is set to TRUE.

Table 5–5 MFPRetail to IP: CP Approved Data

MFP Retail Formula	IP Target Measure
bucpbopc	bucpbopc
bucpbopu	bucpbopu
bucpeopc	bucpeopc
bucpeopu	bucpeopu
bucprecc	bucprecc
bucprecu	bucprecu
bucpbopr	bucpbopr
bucpicogsc	bucpcogsc
bucpeopr	bucpeopr
bucpigmpv	bucpgmpv
bucpmkdpromor + bucpmkdpermr + bucpmkdclrr	bucpmkdr
bucprecr	bucprecr
bucpslsregr + bucpslsclrr + bucpslspror	bucpslsr
bucpslsregu + bucpslsclru + bucpslsprou	bucpslsu

The OP Approved data is sent when mowpappopnewb is set to TRUE.

Table 5–6 MFPRetail to IP: OP Approved Data

MFP Retail Formula	IP Target Measure
buopbopc	buopbopc
buopbopu	buopbopu
buopeopc	buopeopc
buopeopu	buopeopu
buoprecc	buoprecc
buoprecu	buoprecu
buopbopr	buopbopr

Table 5–6 (Cont.) MFPRetail to IP: OP Approved Data

MFP Retail Formula	IP Target Measure
buopicogsc	buopcogsc
buocpeopr	buocpeopr
buopigmpv	buopgmpv
buopmkdpromor + buopmkdpermr + buopmkdclrr	buopmkdr
buoprecr	buoprecr
buopslsregr + buopslsclrr + buopslspror	buopslsr
buopslsregu + buopslsclru + buopslsprou	buopslsu

IP to MFPRetail Package

The following information is about the IP to MFPRetail integration.

Scripts Used By the IP to MFPRetail Package

- ip_to_mfp_init.ksh
- ip_to_mfp_finalize.ksh

Data Mapping for IP to MFPRetail Package

The CP Approved data is sent when ipcpappnewb is set to TRUE.

Table 5–7 IP to MFPRetail: CP Approved Data

IP Formula	MFP Retail Target Measure
ipcpbopc	ipcpbopc
ipcpbopu	ipcpbopu
ipcpeopc	ipcpeopc
ipcpeopu	ipcpeopu
ipcpgmpv	ipcpgmpv
ipcprecc	ipcprecc
ipcprecu	ipcprecu
ipcpslsr	ipcpslsr
ipcpslsu	ipcpslsu
ipcpbopr	ipcpbopr
ipcpeopr	ipcpeopr
ipcprecr	ipcprecr

The OP Approved data is sent when ipopappnewb is set to TRUE.

Table 5–8 IP to MFPRetail: OP Approved Data

IP Formula	MFP Retail Target Measure
ipopbopc	ipopbopc
ipopbopu	ipopbopu
ipopeopc	ipopeopc
ipopeopu	ipopeopu
ipopgmpv	ipopgmpv
ipoprecc	ipoprecc
ipoprecu	ipoprecu
ipopslsr	ipopslsr
ipopslsu	ipopslsu
ipopbopr	ipopbopr
ipopeopr	ipopeopr
ipoprecl	ipoprecl

MFPCost to AP Package

The following information is about the MFPCost to AP integration.

Scripts Used By the MFPCost to AP Package

- mfp_to_ap_init.ksh
- mfp_to_ap_finalize.ksh

Data Mapping for MFPCost to AP Package

Data is sent when mowpappcpnewb is set to TRUE.

Table 5–9 MFPCost to AP Data

MFP Cost Formula	AP Target Measure
bucpgmpv	mfcpposgmr
bucprecc	mfcprecc
bucprecu	mfcprecu
bucpslsu	mfcpslsu
bucpslsr	mfcpslsr

MFPRetail to AP Package

The following information is about the MFPRetail to AP integration.

Scripts Used By the MFPRetail to AP Package

- mfp_to_ap_init.ksh
- mfp_to_ap_finalize.ksh

Data Mapping for MFPRetail to AP Package

Data is sent when mowpappcpnewb is set to TRUE.

Table 5–10 MFPRetail to AP Data

MFP Retail Formula	AP Target Measure
bucprecc	mfcprecc
bucprecu	mfcprecu
bucpagmpv	mfcpposgmr
bucprecr	mfcprecr
bucpslsregr + bucpslsclrr + bucpslspror	mfcpslsr
bucpslsregu + bucpslsclru + bucpslsprou	mfcpslsu

SizeOpt to AP Package

The following information is about the SizeOpt (also known as SPO) to AP integration.

Scripts Used By the SizeOpt to AP Package

- sizeopt_to_ap_init.ksh
- sizeopt_to_ap_finalize.ksh

Data Mapping for SizeOpt to AP Package

Data is sent when populated (not NA).

Table 5–11 SizeOpt to AP Data

SizeOpt Formula	AP Target Measure
sku2sizexxlg	sztyclssszmapb
esprofxxlaxg	sztyarchsp1up
esprofxxlbg	sztyarchsp2up
esprofxxlcxg	sztyarchsp3up
esprofxxldxg	sztyarachsp4up
exportprofxxlg	sztysizeprflup
exportescxxlg	szwpescleveltx
sku2atcdxxlg	szwpssncodeb

AP to IP Package

The following information is about the AP to IP integration.

Scripts Used By the AP to IP Package

- ap_to_ip_init.ksh
- ap_to_ip_finalize.ksh

Data Mapping for AP to IP Package

Data is sent when populated (not NA).

Table 5–12 AP to IP Data

AP Formula	IP Target Measure
bpcpwksclru	bpcpwksclru
bpcpwksregu	bpcpwksregu
bpcsfpcstu	bpcsfpcstu
bpcsfprtlu	bpcsfprtlu
bpcpexitweekdt	bpcpexitweekdt
bpcpstartweekdt	bpcpstartweekdt
bpcsbopc	bpcsbopc
bpcsbopr	bpcsbopr
bpcsbopu	bpcsbopu
bpcseopc	bpcseopc
bpcseopr	bpcseopr
bpcseopu	bpcseopu
bpcsmkdclrr	bpcsmkdclrr
bpcsmkdpermr	bpcsmkdpermr
bpcsmkdpromor	bpcsmkdpromor
bpcslrprcu	bpcslrprcu
bpcspromopr cr	bpcspromopr cr
bpcspempr cr	bpcspempr cr
bpcsrcc	bpcsrcc
bpcsr cr	bpcsr cr
bpcsr cu	bpcsr cu
bpcsslclrc	bpcsslclrc
bpcsslclrr	bpcsslclrr
bpcsslclru	bpcsslclru
bpcpprepacku	bpcpprepacku
bpcsstrcntu	bpcsstrcntu

Hierarchy Integration with ODI

If all of the Fashion Planning Bundle applications are on one server or their domains are accessible from one server (for example, through NFS mounts), you only need to run each hierarchy integration package once. However, if you are integrating applications across different servers, the listed integration steps must be repeated with different contexts and agents once for each server is integrated.

Adding New Products: PROD Hierarchy Integration

Sometimes it is required to add new products/SKUs to the product (PROD) hierarchies. But each application's PROD hierarchy typically contains different dimensions. Therefore, the Fashion Planning Bundle provides an integration package named Broadcast PROD Hierarchy that adds new products while maintaining the hierarchy synchronization between all applications.

To add new products, create a prod.csv file (in CSV format) that defines the following information for each new product:

Table 5–13 Prod.csv File Dimension Positions/Labels

Dimension Position / Label
SKU
SKU_LABEL
SKUP
SKUP_LABEL
SKUG
SKUG_LABEL
CLR
CLR_LABEL
SKP1
SKP1_LABEL
SKP2
SKP2_LABEL
VDRC
VDRC_LABEL
VNDR
VNDR_LABEL
CLGP
CLGP_LABEL
SCLS
SCLS_LABEL
CLASS
CLASS_LABEL
ITGP
ITGP_LABEL

Table 5–13 (Cont.) Prod.csv File Dimension Positions/Labels

Dimension Position / Label
PL1
PL1_LABEL
PL2
PL2_LABEL
DEPT
DEPT LABEL
PGRP
PGRP_LABEL
DVSN
DVSN_LABEL
CMPP
CMPP_LABEL

The dimensions in [Table 5–13](#) are the union or superset of the PROD hierarchies for all Fashion Planning Bundle applications. An example file named prod.csv is provided in the \$RPAS_HOME/doc directory. You can edit a copy of the file using a spreadsheet program or text editor.

Once the new products and their attributes have been added to prod.csv, perform the following steps to add the products to all applications:

1. Place the updated prod.csv file in the \$RPAS_HOME directory.
2. In the Projects pane in Designer, open the **Interfaces** folder, then the **Packages** folder.
3. Double-click **Broadcast PROD Hierarchy**.
4. Select the context/agents appropriate to the application domain locations.

Note that the value of the FPB_APPS environment variable for the agent.sh process determines which domains to update. See the *Oracle Retail ODI Enabled Integration Implementation Guide* for an example of the FPB_APPS setup.

For more information about each application's PROD hierarchy, see that application's user guide.

Adding New Stores: LOC Hierarchy Integration

The procedure for adding new stores to the location (LOC) hierarchy is similar to that of adding new products. The Broadcast LOC Hierarchy package adds new stores, while maintaining the hierarchy synchronization between applications.

To add new stores, create a loc.csv file (in CSV format) that defines the following information for each new store:

Table 5–14 *Loc.csv File Dimension Positions/Labels*

Dimension Position/Label
STR
STR_LABEL
DSTR
DSTR_LABEL
RGN
RGN_LABEL
AREA
AREA_LABEL
CHNL
CHNL_LABEL
CHN
CHN_BAEL
COMP
COMP_LABEL

The dimensions in [Table 5–14](#) are the union or superset of the LOC hierarchies for all Fashion Planning Bundle applications. An example file named loc.csv is provided under the \$RPAS_HOME/doc directory. You can edit a copy of the file using a spreadsheet program or text editor.

Once the new stores and their attributes have been added to loc.csv, perform the following steps to add the stores to all applications:

1. Place the updated loc.csv file in the \$RPAS_HOME directory.
2. In the Projects pane in Designer, open the **Interfaces** folder, then the **Packages** folder.
3. Double-click **Broadcast LOC Hierarchy**.
4. Select the context/agents appropriate to the application domain locations.

Note that the value of the FPB_APPS environment variable for the agent.sh process determines which domains to update. See the *Oracle Retail ODI Enabled Integration Implementation Guide* for an example of the FPB_APPS setup.

For more information about each application's LOC hierarchy, see that application's user guide.

Script Integration

This chapter describes the IP data flow and the script integration of the Fashion Planning Bundle applications.

For information about ODI enabled integration for the Fashion Planning Bundle, see [Chapter 5](#).

Integration Scripts

These scripts can be 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.
- Measure names should not be hard-coded. The names are included in a text file named *<source application>_<destination application>_transferList.txt*. For example, *AP_IP_transferList.txt* includes the measure names exported from AP to IP.

Each transfer file includes two columns that correspond to the following data:

- Measure name
- File name

When the transfer file is used for a load script, it contains the name of the measure to be loaded and the name of the file that contains the data.

For an export script, the transfer file contains the name of the measure in the source application and the name of the ovr file that will be created in the input directory of the destination application. The file names correspond to the measure names in the destination application.

- The scripts have a command line argument to set the maximum number of processes that need to be run in parallel. Setting this argument can help speed up the performance of independent tasks on local domains. The default is 1.
- Do not hard-code domain paths. The paths are entered as command line arguments.

Table 6–1 lists the integration scripts for IP. These scripts are located in `$ITEM_HOME/bin`.

Table 6–1 Integration Scripts

Application	Script Name	Arguments
IP	loadActuals.ksh	measurelist, maxprocesses
MFP	exportToMFP.ksh	maxprocesses, destination, measurelist

Batch Designs

This section contains detailed information on the following integration scripts:

- [Load Actuals Data](#)
- [Export to MFP](#)

Load Actuals Data

Script

loadActuals.ksh

Usage

loadActuals.ksh <measurelist> <maxprocesses>

Table 6–2 Load Actuals Data Usage

Argument	Description	Notes
measurelist	Sets the location of the file which contains the list of measures to be exported.	By default, these files are provided with the package in the <code>\$ITEM_HOME/bin</code> directory. The file name should follow the convention of <code>AP_IP_transferList.txt</code> .
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.

Example

loadActuals.ksh `$ITEM_HOME/bin/AP_IP_transferList.txt` 1

Error Information

Table 6–3 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	One or more arguments are missing.
loadactuals	41	yes	Domain does not exist.
loadactuals	42	no	Data file does not exist.
loadactuals	43	yes	All measure input files are empty or missing.
loadactuals	44	yes	Configuration file does not exist.
loadactuals	45	yes	Errors occurred during the load of one or more measures.

Notes

- This script uses the RPAS loadmeasure utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The script ignores any missing or empty measure load files.
- All measure files are placed into the domain's input folder.
- If there were no errors during the loading of all measures, the input file is archived into the domain's input/processed directory. A date stamp is appended to the end of the measure file name.
- The script does not produce an error when records are rejected from the loaded files. These rejected records are logged in the log output of the load process.

Export to MFP

Script

exportToMFP.ksh

Usage

exportToMFP.ksh <maxprocesses> <destination> <measurelist>

Argument	Description	Notes
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.
destination	Sets the path to the input folder of the destination domain where the exported files are placed.	The export script creates the file containing the exported measures in the input folder of the destination domain.
measurelist	Sets the location of the file which contains the list of measures to be exported.	By default, these files are provided with the package in the \$ITEM_HOME/bin directory. The file name should follow the convention of IP_MFP_transferList.txt.

Example

```
exportToMFP.ksh 1 $MFP_HOME/domain/mfpcst $ITEM_HOME/bin/IP_MFP_transferList.txt
```

Error Information

Table 6–4 Export to MFP Error Information

Task Name	Error Code	Abort Required?	Description of Error
export	200	yes	Argument missing.
export	210	yes	Domain does not exist.
export	220	yes	Missing \$EXPORTLISTFILE.
export	230	no	Errors found in the export log file.

Notes

- This optional script is used only if exports to MFP are required.
- The script uses the RPAS exportData utility to export measure data from the domain. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.

Batch Processing

This chapter contains all the scripts that are needed to run and maintain the IP environment.

Batch Structure Overview

The following directories are used by the batch scripts. These directories are subdirectories of the `$ITEM_HOME` directory.

Table 7-1 *Directories Used by Batch Scripts*

Directory Name	Content of the Directory
bin	Batch scripts
config	IP template configuration
domain	Domains
input	Input files for building the domain
logs	Log files from running any of the batch scripts. A system administrator can scan the logs for any errors, exceptions, or failures. If there are none, the batch completed successfully.
temp	Temporary files used by the batch scripts

Batch Scheduling

Scheduling of the batch scripts are in the following categories:

- [Daily Batch Scripts](#)
- [Weekly Batch Scripts](#)
- [Unscheduled Administration Script](#)

The following information is included in the tables for each batch script:

- A short description of the script
- The name of the script
- The directory in the `$ITEM_HOME/bin` directory where the batch script is found
- Dependencies on other batch scripts

For a detailed description of each script, see the [Batch Designs](#) section in this chapter.

Daily Batch Scripts

The daily batch scripts are run every day before executing the weekly batch scripts.

Table 7–2 lists information on a daily batch scripts.

Table 7–2 Daily Batch Script

Description	Script Name	Batch Directory	Dependency
Backup	N/A	N/A	N/A
Export to MFP	exportToMFP.ksh	bin	Backup
Load Actuals	loadActuals.ksh	bin	Backup

Weekly Batch Scripts

The daily batch script is run before executing the weekly batch scripts.

Table 7–3 lists information on the weekly batch scripts.

Table 7–3 Weekly Batch Scripts

Description	Script Name	Batch Directory	Dependency
Backup	N/A	N/A	N/A
Export to MFP	exportToMFP.ksh	bin	Backup
Formalize DPM Positions	updateDpmPositionStatus.ksh	loadhier	Backup
Calendar Hierarchy Load	loadhier.ksh	loadhier	Backup
Product Hierarchy Load	loadhier.ksh	loadhier	Backup, Formalize DPM Positions (if run)
Location Hierarchy Load	loadhier.ksh	loadhier	Backup
Load on order data	loadActuals.ksh	bin	Calendar hierarchy load, Product hierarchy load, Location hierarchy load
Load MFP data	loadActuals.ksh	bin	Calendar hierarchy load, Product hierarchy load, Location hierarchy load
Load Actuals data	loadActuals.ksh	bin	Calendar hierarchy load, Product hierarchy load, Location hierarchy load
Load AP data	loadActuals.ksh	bin	Calendar hierarchy load, Product hierarchy load, Location hierarchy load
Propagate inventory and aggregate data for all planning levels	processactuals.ksh	actualize	Load on order data, Load MFP data
Generate sales forecast	runforecast.ksh	forecast	Load on order data, Load MFP data

Table 7–3 (Cont.) Weekly Batch Scripts

Description	Script Name	Batch Directory	Dependency
Refresh existing workbooks Note: This script is optional.	refresh.ksh	workbook	Generate sales forecast
Auto build workbooks placed on queue Note: This script is optional.	autobuild.ksh	workbook	None

Unscheduled Administration Script

The following script is not part of a normal batch schedule. This script is executed only to perform the specified activity.

[Table 7–4](#) lists information on the unscheduled administration script.

Table 7–4 Unscheduled Administration Script

Description	Script Name	Batch Directory	Dependency
Load mapping	loadActuals.ksh	bin	None

Batch Environment Scripts

These scripts are included in the other batch scripts to control logging and set environment variables. These batch scripts are only supported for IP. The batch environment scripts supported for IP Configured for COE are described in [Chapter 8](#).

The first script, message.ksh, controls the overall logging. The script writes batch script details to a daily log file. The daily log file is created in the \$ITEM_HOME/logs directory. The format of the file name is MnthID_Day.log, for example, Apr_02.log.

The second script, environment.ksh, is called at the beginning of every batch script. This script sets the following environment variables:

- export ITEM_CONFIGNAME=itemplan
- export ITEM_DOMAINHOME=\$ITEM_HOME/domain
- export ITEM_MASTERDOMAIN=\$ITEM_DOMAINHOME/itemplan
- export ITEM_CONFIGHOME=\$ITEM_HOME/config
- export ITEM_EXPORT=\$ITEM_HOME/export
- export ITEM_INPUThOME=\$ITEM_HOME/input
- export ITEM_LOG_DIR=\$ITEM_HOME/logs
- export ITEM_LIB=\$ITEM_HOME/bin
- export ITEM_TEMP=\$ITEM_HOME/temp
- export ITEM_BATCH=\$ITEM_HOME/bin
- export LOGLEVEL=all

The LOGLEVEL parameter can be set to any of the RPAS supported logging levels—all, profile, debug, audit, information, warning, error, and none.

Batch Designs

This section contains detailed information on the following batch scripts:

- [Auto Build Workbooks](#)
- [Calendar Hierarchy Load](#)
- [Formalization of DPM Positions](#)
- [Product Hierarchy Load](#)
- [Load Actuals Data](#)
- [Load AP Data](#)
- [Load On Order Data](#)
- [Load Actuals Data](#)
- [Load Mapping Measures](#)
- [Location Hierarchy Load](#)
- [Process Actuals Data](#)
- [Refresh Workbooks](#)
- [Run Batch Forecast](#)

Some of the scripts have a command line argument to set the maximum number of processes that need to be run in parallel. Setting this argument can help speed up the performance of independent tasks on local domains. The default is 1.

Formalization of DPM Positions

Script

updateDpmPositionStatus.ksh

Usage

updateDpmPositionStatus.ksh <maxprocesses>

Argument	Description	Notes
maxprocesses	Sets the maximum number of hierarchy load processes to run in parallel.	The default is 1.

Example

updateDpmPositionStatus.ksh 1

Error Information

Table 7–5 Formalization of DPM Positions Error Information

Task Name	Error Code	Abort Required?	Description of Error
updateDpmPositionStatus	90	yes	One or more arguments are missing.
updateDpmPositionStatus	91	yes	Domain does not exist.
updateDpmPositionStatus	92	no	dpmposupdate.xml file does not exist.
updateDpmPositionStatus	93	no	dpmposupdate.xml file is empty.
updateDpmPositionStatus	94	yes	Formalizing DPM positions produced errors. Further batch processing should not be run until the error has been investigated.

Notes

- This script is optional. If no DPM positions are becoming formal, it may be skipped.
- Before running this script, a dpmposupdate.xml file must exist in the input directory of the domain.
- This script uses the RPAS updateDpmPositionStatus utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.

Calendar Hierarchy Load

Script

loadhier.ksh

Usage

loadhier.ksh <hierarchy> <purgeage> <maxprocesses>

Argument	Description	Notes
hierarchy	Sets the RPAS name of the hierarchy to be loaded.	The default for the calendar hierarchy is clnd.
purgeage	Sets the number of days after updating a hierarchy position before purging the position.	The default is 10000.
maxprocesses	Sets the maximum number of hierarchy load processes to run in parallel.	The default is 1.

Example

loadhier.ksh clnd 10000 10

Error Information

Table 7–6 Calendar Hierarchy Load Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadhier	20	yes	One or more arguments are missing.
loadhier	21	yes	Domain does not exist.
loadhier	22	no	Hierarchy file does not exist.
loadhier	24	yes	Loading the hierarchy produced errors. Further batch processing should not be run until the error has been investigated.

Notes

- Before running this script, a clnd.dat file must exist in the input directory of the domain.
- This script uses the RPAS loadHier utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The calendar, product, and location hierarchy loads can be run in any order.

Product Hierarchy Load

Script

loadhier.ksh

Usage

loadhier.ksh <hierarchy> <purgeage> <maxprocesses>

Argument	Description	Notes
hierarchy	Sets the RPAS name of the hierarchy to be loaded.	The default for the product hierarchy is prod.
purgeage	Sets the number of days after updating a hierarchy position before purging the position.	The default is 10000.
maxprocesses	Sets the maximum number of hierarchy load processes to run in parallel.	The default is 1.

Example

loadhier.ksh prod 10000 10

Error Information

Table 7–7 Product Hierarchy Load Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadhier	20	yes	One or more arguments are missing.
loadhier	21	yes	Domain does not exist.
loadhier	22	no	Hierarchy file does not exist.
loadhier	24	yes	Loading the hierarchy produced errors. Further batch processing should not be run until the error has been investigated.

Notes

- Before running this script, a prod.dat file must exist in the input directory of the domain.
- This script uses the RPAS loadHier utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The calendar, product, and location hierarchy loads can be run in any order.

Location Hierarchy Load

Script

loadhier.ksh

Usage

loadhier.ksh <hierarchy> <purgeage> <maxprocesses>

Argument	Description	Notes
hierarchy	Sets the RPAS name of the hierarchy to be loaded.	The default for the product hierarchy is loc.
purgeage	Sets the number of days after updating a hierarchy position before purging the position.	The default is 10000.
maxprocesses	Sets the maximum number of hierarchy load processes to run in parallel.	The default is 1.

Example

loadhier.ksh loc 10000 10

Error Information

Table 7–8 Location Hierarchy Load Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadhier	20	yes	One or more arguments are missing.
loadhier	21	yes	Domain does not exist.
loadhier	22	no	Hierarchy file does not exist.
loadhier	24	yes	Loading the hierarchy produced errors. Further batch processing should not be run until the error has been investigated.

Notes

- Before running this script, a loc.dat file must exist in the input directory of the domain.
- This script uses the RPAS loadHier utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The calendar, product, and location hierarchy loads can be run in any order.

Load On Order Data

Script

loadActuals.ksh

Usage

loadActuals.ksh <measurelist> <maxprocesses>

Argument	Description	Notes
measurelist	Sets the location of the file which contains the list of measures to be exported.	By default, these files are provided with the package in the \$ITEM_HOME/bin directory. The file name should follow the convention of AP_IP_transferList.txt.
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.

Control File

LoadOnOrderList.txt: Contains the following list of measures that can be loaded as part of this script:

- ipwponordc
- ipwponordr
- ipwponordu

Example

loadActuals.ksh LoadOnOrderList.txt 1

Error Information

Table 7–9 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	One or more arguments are missing.
loadactuals	41	yes	Domain does not exist.
loadactuals	42	no	Data file does not exist.
loadactuals	43	yes	All measure input files are empty or missing.
loadactuals	44	yes	Configuration file does not exist.
loadactuals	45	yes	Errors occurred during the load of one or more measures.

Notes

- This script uses the RPAS loadmeasure utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The script ignores any missing or empty measure load files.
- All measure files are placed into the domain's input folder.
- If there were no errors during the loading of all measures, the input file is archived into the domain's input/processed directory. A date stamp is appended to the end of the measure file name.
- The script does not produce an error when records are rejected from the loaded files. These rejected records are logged in the log output of the load process.

Load Actuals Data

Script

loadActuals.ksh

Usage

loadActuals.ksh <measurelist> <maxprocesses>

Argument	Description	Notes
measurelist	Sets the location of the file which contains the list of measures to be exported.	By default, these files are provided with the package in the \$ITEM_HOME/bin directory. The file name should follow the convention of AP_IP_transferList.txt.
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.

Control File

LoadActualsList.txt: Contains the following list of measures that can be loaded as part of this script:

- drtybopclrc
- drtybopclrr
- drtybopclru
- drtybopregc
- drtybopregr
- drtybopregu
- drtyeopclrc
- drtyeopclrr
- drtyeopclru
- drtyeopregc
- drtyeopregr
- drtyeopregu
- drtymkdar
- drtyrecc
- drtyrecr
- drtyrecu
- drtyslclrc
- drtyslclrr
- drtyslclru
- drtyslproc
- drtyslpror
- drtyslprou
- drtyslregc

- drtyslsregr
- drtyslsregu

Example

loadActuals.ksh LoadActualsList.txt 1

Error Information

Table 7–10 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	One or more arguments are missing.
loadactuals	41	yes	Domain does not exist.
loadactuals	42	no	Data file does not exist.
loadactuals	43	yes	All measure input files are empty or missing.
loadactuals	44	yes	Configuration file does not exist.
loadactuals	45	yes	Errors occurred during the load of one or more measures.

Notes

- This script uses the RPAS loadmeasure utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The script ignores any missing or empty measure load files.
- All measure files are placed into the domain's input folder.
- If there were no errors during the loading of all measures, the input file is archived into the domain's input/processed directory. A date stamp is appended to the end of the measure file name.
- The script does not produce an error when records are rejected from the loaded files. These rejected records are logged in the log output of the load process.

Load AP Data

Script

loadActuals.ksh

Usage

loadActuals.ksh <measurelist> <maxprocesses>

Argument	Description	Notes
measurelist	Sets the location of the file which contains the list of measures to be exported.	By default, these files are provided with the package in the \$ITEM_HOME/bin directory. The file name should follow the convention of AP_IP_transferList.txt.
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.

Control File

AP_IP_transferList.txt: Contains the following list of measures that can be loaded as part of this script:

- apcpbopc
- apcpbopr
- apcpbopu
- apcpcojsc
- apcpeopc
- apcpeopr
- apcpeopu
- apcpexitd
- apcpmkdclrr
- apcpmkdiniop
- apcpmkdpermr
- apcpmkdpror
- apcppckszu
- apcpprcclrcp
- apcpprcclrr
- apcpprccp
- apcpprcinir
- apcpprcprocp
- apcpprcpror
- apcpprcr
- apcprecc
- apcprecr
- apcprecu

- apcpstcp
- apcpslsclrc
- apcpslsclrr
- apcpslsclru
- apcpslsregc
- apcpslsregr
- apcpslsregu
- apcpslsstartd
- apcpstru
- apcpweeksclru
- apcpweeksregu

Example

loadActuals.ksh AP_IP_transferList.txt 1

Error Information

Table 7–11 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	One or more arguments are missing.
loadactuals	41	yes	Domain does not exist.
loadactuals	42	no	Data file does not exist.
loadactuals	43	yes	All measure input files are empty or missing.
loadactuals	44	yes	Configuration file does not exist.
loadactuals	45	yes	Errors occurred during the load of one or more measures.

Notes

- This script uses the RPAS loadmeasure utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The script ignores any missing or empty measure load files.
- All measure files are placed into the domain's input folder.
- If there were no errors during the loading of all measures, the input file is archived into the domain's input/processed directory. A date stamp is appended to the end of the measure file name.
- The script does not produce an error when records are rejected from the loaded files. These rejected records are logged in the log output of the load process.

Process Actuals Data

Script

processactuals.ksh

Usage

processactuals.ksh

Error Information**Table 7-12 Process Actuals Error Information**

Task Name	Error Code	Abort Required?	Description of Error
processactuals	60	yes	Domain does not exist.
processactuals	61	yes	Errors occurred during the processing of actualized data.

Notes

- This script uses the RPAS mace utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The script can run the rule groups batch_calc, batch_inv_adj, batch_inv_roll1, and batch_inv_roll2.

Run Batch Forecast

Script

runforecast.ksh

Usage

runforecast.ksh

Error Information

Table 7–13 Run Batch Forecast Error Information

Task Name	Error Code	Abort Required?	Description of Error
runforecast	70	yes	Domain does not exist.
runforecast	71	yes	Errors occurred during forecast creation. Only forecast measures are affected by this.

Notes

- This script uses the RPAS mace utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The script can run the batch_forecast rule group. This rule group can execute the In-Season and Pre-Season embedded forecasts.

Refresh Workbooks

Script

refresh.ksh

Usage

refresh.ksh <maxprocesses>

Argument	Description	Notes
maxprocesses	Sets the maximum number of refresh processes to run in parallel.	The default is 1.

Example

refresh.ksh 10

Error Information

Table 7–14 Refresh Workbooks Error Information

Task Name	Error Code	Abort Required?	Description of Error
refresh	120	yes	Argument missing.
refresh	121	yes	Domain does not exist.
refresh	122	no	Workbook list is either blank or missing. This may be the result of there being no workbooks available to refresh.
refresh	123	no	Errors occurred during workbook refresh processing.

Notes

- This is an optional script. It enables all workbooks to be kept current with the elapsed settings and prevents stale data from being committed.
- The script uses the RPAS wbbatch utility to refresh workbooks in the auto build queue. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- Workbooks should be committed before the refresh batch job. If a workbook is not committed, the data in that workbook can be overwritten by data in the domain during the refresh process.
- If you are running both the auto build and refresh scripts, it is important to run the refresh job first. The refresh updates all IP workbooks saved in the domain. If the auto build is run first, these workbooks are unnecessarily refreshed.

Auto Build Workbooks**Script**

autobuild.ksh

Usage

autobuild.ksh <maxprocesses>

Argument	Description	Notes
maxprocesses	Sets the maximum number of autobuild workbook processes to run in parallel.	The default is 1.

Example

autobuild.ksh 10

Error Information**Table 7–15 Auto Build Workbooks Error Information**

Task Name	Error Code	Abort Required?	Description of Error
autobuild	110	yes	Argument missing.
autobuild	111	yes	Domain does not exist.
autobuild	112	no	Errors occurred during auto workbook build processing.

Notes

- This script is an optional script. It should be used only if auto workbooks are utilized.
- The script uses the RPAS wbbatch utility to build workbooks in the auto build queue. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- Auto build entries must be added via the domain before running this script. See the *Oracle Retail Predictive Application Server User Guide* for details on this process.
- If you are running both the auto build and refresh scripts, it is important to run the refresh job first. The refresh updates all IP workbooks saved in the domain. If the auto build is run first, these workbooks are unnecessarily refreshed.

Load Mapping Measures

Script

loadActuals.ksh

Usage

loadActuals.ksh <measurelist> <maxprocesses>

Argument	Description	Notes
measurelist	Sets the location of the file which contains the list of measures to be exported.	By default, these files are provided with the package in the \$ITEM_HOME/bin directory. The file name should follow the convention of AP_IP_transferList.txt.
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.

Control File

LoadMapList.txt: Contains the list of measures that can be loaded as part of this script.

Example

loadActuals.ksh LoadMapList.txt 1

Error Information

Table 7–16 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	One or more arguments are missing.
loadactuals	41	yes	Domain does not exist.
loadactuals	42	no	Data file does not exist.
loadactuals	43	yes	All measure input files are empty or missing.
loadactuals	44	yes	Configuration file does not exist.
loadactuals	45	yes	Errors occurred during the load of one or more measures.

Notes

- This script uses the RPAS loadmeasure utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The script ignores any missing or empty measure load files.
- All measure files are placed into the domain's input folder.
- If there were no errors during the loading of all measures, the input file is archived into the domain's input/processed directory. A date stamp is appended to the end of the measure file name.
- The script does not produce an error when records are rejected from the loaded files. These rejected records are logged in the log output of the load process.

Item Planning Configured for COE

This chapter provides information on integration and batch processing specifically for IP when it is configured for COE.

For information on installing COE, see the Oracle Retail IP Configured for COE documentation.

Integration Methods and Communication Flow

This section describes the flow of data between IP Configured for COE and COE. The retailer can define one server for both COE and IP/RPAS or define separate servers.

Data Exported from COE to IP/RPAS

Optimized forecasts are extracted from COE into pipe-delimited flat files. The data is extracted at the optimization level, which is the style-color/price zone level. The input files are not in RPAS load-ready format and need to be transformed as explained below for each file.

The following data is exported from COE to IP/RPAS.

- [Forecast Summaries](#)
- [Forecast Activities](#)
- [Markdown Recommendations](#)

Note: IP does not have a pricing group.

Forecast Summaries

The forecast summaries file, mdo-rpas-forecast-summ.txt, contains forecasted opportunity costs for the individual style-colors or price groups. [Table 8–1](#) describes the layout of this file.

Table 8–1 *Layout of the Forecast Summaries File*

Field Name	Data Type	Maximum Length	Nullable?	Field Description
Merchandise Key	String	25	No	Client merchandise identifier in the product hierarchy
Location Key	String	25	No	Client location identified in the location hierarchy

Table 8–1 (Cont.) Layout of the Forecast Summaries File

Field Name	Data Type	Maximum Length	Nullable?	Field Description
Item Opportunity Cost	Decimal	22,2	Yes	Forecasted opportunity cost as an item
Pricing Group Opportunity Cost	Decimal	22,2	Yes	Forecasted opportunity cost as a member of a pricing group
Item Full Price	Decimal	22,2	Yes	Original item retail price as configured in COE

The following is an example of the content of the Forecast Summaries file:

```
10104005429885492          | 711          | 0.00
| 0.00          | 48.00

10104005429885492          | 713          | 0.00
| 0.00          | 48.00
```

In order to load this file into RPAS, the file needs to be transformed into a CSV file. Since all of the measures are not going to be loaded into the IP domain, the transformation is more than just replacing the pipe delimiter with the comma delimiter. Each measure listed above is put into a separate CSV file in the format of merchandise key, location key, measure value. [Table 8–2](#) provides details on the file.

Table 8–2 Forecast Summaries File After Transformation

Measure Name	Measure Label	File Name
IPWPCORCOPCOV (COE Recommended Opportunity Cost (measure 94 - Measure Analysis))	Item Opportunity Cost	coeitemoppcost.csv.rpl

The following is an example of the content of the file after transformation:

```
coeitemoppcost.csv.rpl
10104005429885492,711,0.00
10104005429885492,713,0.00
```

Forecast Activities

The forecast activities file, mdo-rpas-forecast-act.txt, contains forecast units and prices by fiscal week for the style-color/price zone as an individual item or as part of a pricing group. [Table 8–3](#) describes the layout of this file.

Table 8–3 Layout of the Forecast Activities File

Field Name	Data Type	Maximum Length	Nullable?	Field Description
Merchandise Key	String	25	No	Client merchandise identifier in the product hierarchy
Location Key	String	25	No	Client location identified in the location hierarchy
Forecast Date	Date in format YYYY-MM-DD	10	No	Start date of the fiscal week for which forecast is applicable
Forecasted Item Sales Units	Integer	22	Yes	Forecasted sales units for the time period

Table 8–3 (Cont.) Layout of the Forecast Activities File

Field Name	Data Type	Maximum Length	Nullable?	Field Description
Forecasted Item Sales Price	Decimal	22,2	Yes	Forecasted price per unit for the time period
Forecasted Item Inventory Units	Integer	22	Yes	Forecasted inventory at the end of the time period
Forecasted Item Ticket Price	Decimal	22,2	Yes	Forecasted ticket price per unit for the time period
Forecasted Pricing Group Sales Units	Integer	22	Yes	Forecasted sales units for the time period as member of a pricing group
Forecasted Pricing Group Sales Price	Decimal	22,2	Yes	Forecasted price per unit for the time period as member of a pricing group
Forecasted Pricing Group Inventory Units	Integer	22	Yes	Forecasted inventory at the end of the time period as member of a pricing group
Forecasted Pricing Group Ticket Price	Decimal	22,2	Yes	Forecasted ticket price per unit for the time period as member of a pricing group

The following is an example of the content of the Forecast Activities file:

```

10100004769639001          | 629          | 2004-10-09 | 1
| 89.97          | 0          | 149.95          | 1          | 89.97          | 0
| 149.95

10100004769639001          | 629          | 2004-10-16 | 0
| 149.95          | 0          | 149.95          | 0          | 149.95          | 0
| 149.95

```

In order to load this file into RPAS, the file needs to be transformed into a CSV file. The transformation includes ordering of the hierarchy information, formatting of the calendar position and creating one file per measure. The order of the hierarchy positions should be calendar, product, and then location. The calendar position in the export file is the week ending date. This also corresponds to the week position in the calendar hierarchy. The export format is YYYY-MM-DD and is transformed to the format of YYYYMMDD in the input file.

Table 8–4 provides details on the files.

Table 8–4 Forecast Activities File After Transformation

Measure Name	Measure Label	File Name
IpWpCoRcFcSULV	Forecasted Item Sales Units	coeitmslsu.csv.rpl
IpWpCoPrSlsArV	Forecasted Item Sales Price	coeitmslspc.csv.rpl
IPWPCOPEOPIUV	Forecasted Item Inventory Units	coeitminvu.csv.rpl
IPWPCORCMDPPV	Forecasted Item Ticket Price	coeitmtkdprc.csv.rpl

The following is an example of the content of the file after transformation:

```
coeitmslsu.csv.rpl
20041009,10100004769639001,629,1
20041016,10100004769639001,629,0

coeitmslsprc.csv.rpl
20041009,10100004769639001,629,89.97
20041016,10100004769639001,629,149.95

coeitminvu.csv.rpl
20041009,10100004769639001,629,0
20041016,10100004769639001,629,0

coeitmtkdprc.csv.rpl
20041009,10100004769639001,629,149.95
20041016,10100004769639001,629,149.95
```

Markdown Recommendations

The recommended markdowns file, mdo-rpas-markdown-act.txt, markdown pricing recommendation by start date for the merchandise-location as an individual item or as part of a pricing group. [Table 8–5](#) describes the layout of this file.

Table 8–5 Layout of the Markdown Recommendations File

Field Name	Data Type	Maximum Length	Nullable?	Field Description
Merchandise Key	String	25	No	Client merchandise identifier in the product hierarchy
Location Key	String	25	No	Client location identified in the location hierarchy
Markdown Date	Date in format YYYY-MM-DD	10	No	Recommended start date of markdown
Item Price Type	String	2	No	Recommended markdown price type as member of a pricing group
Item Markdown Price	Decimal	22,2	Yes	Recommended markdown (ticket) price
Item Percent Off	Decimal	3,2	Yes	Percent of previous time period's price
Pricing Group Price Type	String	2	No	Recommended markdown price type as member of a pricing group
Pricing Group Markdown Price	Decimal	22,2	Yes	Recommended markdown ticket price as member of a pricing group
Pricing Group Percent Off	Decimal	3,2	Yes	Percent of previous time period's ticket price as member of a pricing group

The following is an example of the content of the Markdown Recommendations file:

```
10104005429885212 | 102
| 2004-10-11 | PP | 36.00 | 0.25 | |
10104005429885212 | 102 | 2004-11-15 | |
| | PP | 31.20 | 0.35
```

In order to load this file into RPAS, the file needs to be transformed into a CSV file. The transformation includes ordering of the hierarchy information, formatting of the calendar position and creating one file per measure. The order of the hierarchy positions should be calendar, product, and then location. The calendar position in the export file is a specific date. This corresponds to a specific day position in the calendar hierarchy. The export format is YYYY-MM-DD and is transformed to the format of YYYYMMDD in the input file. Since the measures are based at week in IP, the day level information is aggregated during the load process. [Table 8–6](#) provides details on the files.

In the example, it appears that certain values are blank. In these cases, the output file will not contain a record for the corresponding week/style-color/price zone that has blank input.

Table 8–6 Markdown Recommendations File After Transformation

Measure Name	Measure Label	File Name	Notes
IpWpCoeRCMdPDV	Item Markdown Price	coeitmmkdprc.csv.rpl	Aggregation Type of average_pop

The following is an example of the content of the file after transformation:

```
coeitmmkdprc.csv.rpl
20041011,10104005429885212,102,36.00
```

Data Exported from IP/RPAS to COE

Business rules are exported from IP/RPAS to COE.

Business Rules

COE supports the setting of business rules that drive the optimization process. Within the IP environment, three of the business rules are set and eventually exported to COE for optimization.

The following business rules are used with COE. In these rules, EOL means end of life.

- [COE Initial EOL Salvage Value % Off](#)
- [COE Initial EOL Exit Week](#)
- [COE Initial EOL Sell Thru %](#)

All business rules are managed and maintained in COE through the COE Business Rule Manager screen.

COE Initial EOL Salvage Value % Off The percent off the current ticketed retail price that is used to determine the salvage value price for any remaining inventory at the exit date.

COE Initial EOL Exit Week Also known as Out Date, this is the date when the retailer no longer wants to sell the item or carry it in stores.

COE Initial EOL Sell Thru % This is the desired percentage of the total inventory to be sold at the EOL exit week (which calculates ending inventory).

Example This is an example of using the business rules.

Item 123456

COE Initial EOL Salvage Value % Off = 100%

COE Initial EOL Exit Week = 02/20/09

COE Initial EOL Sell Thru % = 95%

The goal is to sell 95% of the inventory by February 20, 2009. In order to maintain the best possible gross margin, a weekly forecast and recommended markdowns for the item are obtained from COE. The initial business rules are needed. These are the solid driving factors, or goals, to produce a forecast and markdown recommendations.

Format of the Export File

The business rules are set in the workbook for IP Configured for COE. The measures for the rules are set at the Style/Color/Price Zone level. They have to be set in order for the initial forecasts to run within COE. The three measures are exported for import into COE.

The first step is to translate the COE Initial EOL Exit Week to a week ending date. The value that is stored for this measure is the position ID for the week that was selected. An internal measure is required to support this translation. The new internal measure (Initial EOL Exit Date) is a date measure. A rule in the workbook will use the End of Week Date measure to translate the position ID for the week in the COE Initial EOL Exit Week measure into a date in the Initial EOL Exit Date measure. The following is the expression for the date measure:

lookup(End of Week Date, [cldn].[week], COE Initial EOL Exit Week)

The second step is to extract an individual file for each measure (Initial EOL Exit Date, COE Initial EOL Sell Thru %, and COE Initial EOL Salvage Value % Off) in csv format. The date format needs to be YYYY-MM-DD format. The output is in the following format:

STYLE_COLOR_ID,PRICE_ZONE_ID,MEASURE_VALUE

The following is an example for the Initial Exit Date file:

10120012373841242,60,2002-01-01

This output needs to be translated into a loadable format for COE. [Table 8–7](#) describes the specification for the ash_brm_instance_tbl.dat COE file.

Table 8–7 COE Load File Description

Attribute	Type	Maximum Length	Nullable	Description
Merchandise_Key	String	50	No	Key for this level of the hierarchy
Merchandise_Level	String	50	No	ID for this level of the hierarchy
Location_Key	String	50	No	Key for this level of the hierarchy
Location_Level	String	50	No	ID for this level of the hierarchy
Rule_Name	String	64	No	The name of the business rule associated with the item
Rule_Value	String Note: Values less than one are expressed as 0.n	100	No	The business rule value assigned to the item
Attrib1_Value	String	100	Yes	The specific value associated with the item for custom attribute 1
Attrib2_Value	String	100	Yes	The specific value associated with the item for custom attribute 2
Delete_Flag	Integer	1	No	A flag to indicate whether the instance is to be deleted or inserted 0 = insert (the default) 1 = delete

The following is an example of the COE load file:

```
10120012373841242|PRODUCT KEY|60|STORE|PLANNED_START_
DT|2002-01-01|||0|
10120012373841305|PRODUCT KEY|63|STORE|PLANNED_START_
DT|2002-01-01|||0|
10120012373841305|PRODUCT KEY|60|STORE|PLANNED_START_
DT|2002-01-01|||0|
```

For the measures exported from IP, the following shows the definition of the Rule_Name:

- OUT_DT = Initial EOL Exit Date
- INVENTORY_TARGET = COE Initial EOL Sell Thru %
- SALVAGE_ABOVE_TARGET = COE Initial EOL Salvage Value % Off

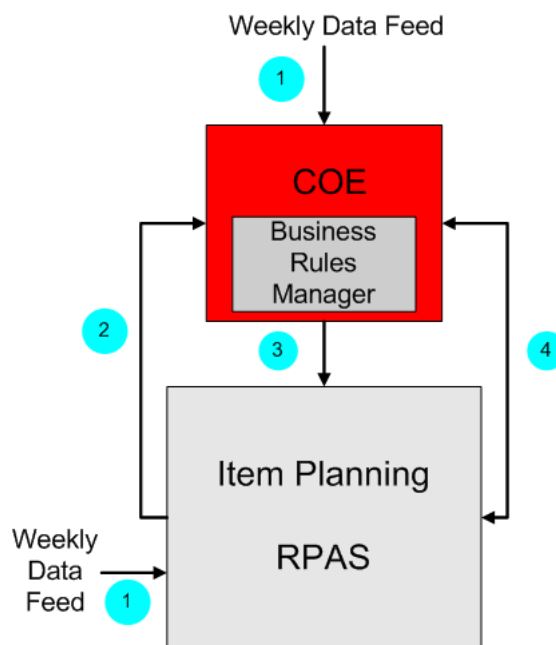
Each exported file needs to be translated into the load format as shown above. In summary, the following are the load file requirements:

- The file must become pipe (|) delimited.
- The first field will be equal to the style-color ID that is exported.
- The second field will be equal to style-color.
- The third field will be equal to the price zone ID that is exported.
- The fourth field will be equal to price zone.
- The fifth field will be equal to the corresponding Rule_Name.
- The sixth field will be the measure value that is exported.
- The seventh and eighth fields will be null at this point.
- The ninth field will be set to zero.

Data Flow Between COE and IP/RPAS

Figure 8–1 illustrates the flow of data between COE and IP/RPAS. For an explanation of each numbered item on the diagram, see the [Data Flow Description](#) below.

Figure 8–1 Data Flow Between COE and IP/RPAS



Data Flow Description

The following scripts are provided:

- RPAS provides the `rpas2coe.sh` script for exporting and transforming data sent to COE.
- RPAS provides the `coe2rpas.sh` script to transform and load COE extracts to IP. Retailers are responsible for making sure the COE extracted files are placed in the `$COE_HOME/bin/fromMDO` directory.

The following steps explain the data flow:

1. On a weekly basis, the retailer provides historical sales data and hierarchy updates to COE and IP/RPAS. The weekly feeds into COE and IP/RPAS are not necessarily the same. The retailer generates the files to provide the data and saves the files in the correct location for each server. A schedule for the data feed needs to be set up.
2. COE is responsible for loading the business rules extracts from IP/RPAS on a nightly or weekly basis. Transformation of the files is done.
3. IP/RPAS has the responsibility to load the data from COE. It expects files in be in a specific directory. The retailer defines the directory and places any necessary files in that directory. The data sent from COE to IP/RPAS is not historical or transactional. It is forecast information obtained by running optimization, that is, forecasted sales, inventory, and markdowns.

For information on whether a business rule can be edited, see the Oracle Retail Clearance Optimization Engine documentation..

The extract from COE is not loadable. Transformation of the file only involves minor changes like reorganizing fields and changing separators.

4. Within IP/RPAS, a user can enter in What-If information. This information is sent to COE via the COE special expression for optimization. COE will return revised optimization information like forecasted sales, inventory, and markdowns. This is all real-time interaction.

RPAS Special Expression for COE

The communication between IP and COE is accomplished through the use of an RPAS special expression. This special expression sends data from RPAS to COE for optimization. COE returns the data back to RPAS for review in the IP process.

Overview on Evaluation of the Expression by RPAS

The expression is implemented in Java using the RPAS Java Special Expression API. This special expression is not a part of the core RPAS functionality, but is layered on top of RPAS. It behaves, from the point of view of the RPAS Calculation Engine, in a manner identical to that of other special expressions.

When data is imported from COE into RPAS, either as part of a workbook load or refresh or as part of the execution of a custom menu, the RPAS Calculation Engine evaluates the COE special expression. The COE expression object provides the values, some configured and some specified at run time through measure values, necessary to initiate and perform a transaction with the COE data source.

After the required information has been passed to the RPAS Java Special Expression API, the COE expression object initiates a transaction with the COE service web service. Communication between RPAS and the COE service is encoded as an XML message.

After the COE service has received the request, it unpacks the communication and uses the request information to retrieve the desired data from the COE application. This data is encoded in a response that is returned to the COE expression that, in turn, supplies the data to the RPAS Calculation Engine.

Environment Requirements

The following environment requirements should be considered:

- [Configuration of the MDO_URL Measure](#)
- [Security Features](#)
- [COE Expression Runtime Environment](#)

Configuration of the MDO_URL Measure

In the input arguments, svcurl specifies the measure that holds the COE Service URL string. This measure should be called MDO_URL to be consistent with other RPAS meta measure names. It should be configured in the domain configuration during implementation. The measure must be scalar with a persistent database and its value type must be string.

The COE Service URL string that this measure stores should be fairly static and not accessible to an end user. It is recommended that the basestate and writestate are set to read-only and that insertable is set to false.

The MDO_URL measure must be included in any workbook that invokes the COE expression. Workbooks that have this measure may not include it in any worksheet because the measure should not be viewed by the end user. However, the workbook's load rule group and refresh rule group must include a rule like the following so that the measure can be loaded correctly into the workbook:

```
MDO_URL=MDO_URL.master
```

There is no need to create a calc rule and commit rule for this measure. To populate the MDO_URL initially with the correct COE Service URL string, a system administrator can run a mace call from a console window after the domain is built:

```
mace -d [PathToDomain] -run -expression "MDO_URL=\"XXX\""
```

In the above expression, XXX is the COE Service URL. All other arguments are values defined within the COE application.

Security Features

The COE special expression implements the WS-Security standard for authentication, encryption, and authorization of webservice calls made from RPAS to COE. The implementation is specific to the application server on which the COE web service is running. The special expression supports the Oracle Application Server (OAS) and the Oracle WebLogic Server (OWL). The IP environment setup will be slightly different depending on the application server on which the webservice is running.

Setup for Secure Calls to the Oracle WebLogic Server

The security policy in the Oracle WebLogic Server (OWL) is specified as part of the webservice schema. The client supports several security scenarios which are described below. The client supports Java keystore for storing private keys and trusted certificates. The Sun keytool utility can be used to maintain keystores and generate keys and certificates. The special expression will need the location and admin credentials of the keystore. Two additional measures, keystore_file and keystore_adm, are configured in the RPAS domain to achieve this. These are string scalar measures and have the same configuration properties as that of the MDO_URL measure.

■ Keystore_file Measure

This measure stores the location of the Java keystore file. Run the following mace command to set the measure value to keystore location:

```
mace -d [PathToDomain] -run -expression "keystore_file=\"XXX\""
```

In the above expression, XXX is the location of the keystore file. All other arguments are values defined within the COE application.

■ Keystore_adm Measure

This measure stores the username for the keystore admin account. To configure the keystore admin user in RPAS domain, perform the following:

1. Use RPAS usermgr utility to create a user account:

```
usermgr -d [PathToDomain] -add keyadm -label keyadm -password keyadm_password -group keyadm_group
```

Replace the credentials in italics with actual username and password. You can choose any legal username, however the password must be the same as the keystore password.

2. Populate the keystore_adm measure with the newly created username:

```
mace -d [PathToDomain] -run -expression "keystore_adm=\"keyadm\""
```

In the above expression replace keyadm with the actual username.

Authentication An IPCOE system user account will be created on both RPAS IP domain and COE webserver. The account will have the same username and password credentials in both places. All the webservice calls made from IP will pass the IPCOE system user credentials in the security header. It is advisable to encrypt the security header. The system user account is used to avoid the overhead of syncing multiple accounts on COE and RPAS. This will not be needed after the single sign-on functionality is implemented.

To configure the IP/COE system user in RPAS domain, perform the following:

1. Use RPAS usermgr utility to create a system user:

```
usermgr -d [PathToDomain] -add coe_system_user -label coe_system_user_label -password system_user_password -group coe_system_user_group
```

Replace the credentials in italics with actual username and password. The username and password should be identical to the ones configured on COE side.

2. Populate the mdo_suser measure with the newly created username:

```
mace -d [PathToDomain] -run -expression "mdo_suser=\"coe_system_user\""
```

In the above expression replace *coe_system_user* with the actual username. The configuration properties of the mdo_suser measure are identical to the MDO_URL measure mentioned earlier.

Confidentiality The confidentiality policy is configured through the WSDL. The security administrator can configure encryption and keywrapping algorithms to be used for encryption. The message parts that need to be encrypted can also be configured. X509 certificates are used for encryption. The measure *keyalias_enc* needs to be configured in IP. This measure stores the keystore alias that will be used for providing certificates for client side encryption. The alias must be registered in the keystore specified by the *keystore_file* measure.

To configure *keyalias_enc* user in RPAS domain, perform the following:

1. Use RPAS *usermgr* utility to create a user account:

```
usermgr -d [PathToDomain] -add keyalias_enc -label keyalias_enc -password  
keyalias_password -group keyalias_group
```

Replace the credentials in italics with actual aliasname and password stored in the keystore. The password for the alias should be identical in keystore and RPAS.

2. Populate the *keyalias_enc* measure with the newly created username:

```
mace -d [PathToDomain] -run -expression "keyalias_enc=\"keyalias_enc\""
```

In the above expression replace *keyalias_enc* with the actual username.

Integrity The identity policy is also configured through the WSDL. The security administrator can configure signature and digest algorithms and message parts to be signed. X509 certificates are used for signing. A measure *keyalias_sig* needs to be configured in IP. This measure stores the keystore alias that will be used for providing certificates for client side signing. The alias must be registered in the keystore specified by the *keystore_file* measure.

To configure *keyalias_sig* user in RPAS domain, perform the following:

1. Use RPAS *usermgr* utility to create a user account :

```
usermgr -d [PathToDomain] -add keyalias_sig -label keyalias_sig -password  
keyalias_password -group keyalias_group
```

Replace the credentials in italics with actual aliasname and password stored in the keystore. The password for the alias should be identical in keystore and RPAS.

2. Populate the *keyalias_sig* measure with the newly created username:

```
mace -d [PathToDomain] -run -expression "keyalias_sig=\"keyalias_sig\""
```

In the above expression replace *keyalias_sig* with the actual username.

Setup for Secure Calls to Oracle Application Server

The security configuration for the Oracle Application Server (OAS) client is configured through the `ClearanceOptimizationSoapHttp_Stub.xml` file. This file is packaged under the `coeexpression_oas.jar` file located under `$COE_HOME/java/lib` folder. This file needs to be in sync with the configuration specified at the server side. As opposed to the OWL configuration, the location of keystore, keystore password, encryption, and signature alias and their passwords are all configured through this file.

Authentication An IPCOE system user account will be created on both RPAS IP domain and COE webserver. The account will have the same username and password credentials in both places. All the webservice calls made from IP will pass the IPCOE system user credentials in the security header. It is advisable to encrypt at least the security header. The system user account is used to avoid the overhead of syncing multiple accounts on COE and RPAS. This will not be needed after single sign on functionality is implemented.

To configure IP/COE system user in RPAS domain:

1. Use RPAS `usermgr` utility to create a system user:

```
usermgr -d [PathToDomain] -add coe_system_user -label coe_system_user_label
-passwd system_user_password -group coe_system_user_group
```

Replace the credentials in italics with actual username and password. The username and password should be identical to the ones configured on COE side.

2. Populate the `mdo_suser` measure with the newly created username. The configuration properties of the `mdo_suser` measure are identical to the `mdo_url` measure mentioned earlier.

```
mace -d [PathToDomain] -run -expression "mdo_suser=\"coe_system_user\""
```

In the above expression replace *coe_system_user* with the actual username.

Auditing The username for the logged-in user is attached as a custom header to the webservice call. The webservice extracts the username and logs it for auditing. If the expression is invoked outside a workbook, an exception is returned since there is no logged-in RPAS user.

The signature of the special expression is same for OWL and OAS; however, the OAS special expression ignores `keystore_file`, `keystore_adm`, `keyalias_enc`, and `keyalias_sig` measures, since these strings are provided from the `ClearanceOptimizationSoapHttp_Stub.xml` file.

COE Expression Runtime Environment

The COE special expression extends the RPAS Java Expression class. The RpasJavaExpression library should be registered with the domain at build or patch time.

The RPAS_JAVA_CLASSPATH environment variable is used to provide the classpath for the classes used by the java special expression.

- Calls to OAS : This requires path to the following jar files in the given order:
 - oracleRpasUtils.jar located under \$RPAS_HOME/lib
 - wsclient_extended_101320.jar located under \$COE_HOME/java/lib
 - coeexpression_oas.jar located under \$COE_HOME/java/lib

This can be set on Unix as follows:

```
export RPAS_JAVA_CLASS_PATH="$RPAS_HOME/lib/oracleRpasUtils.jar:$COE_
HOME/java/lib/wsclient_extended_101320..jar:$COE_HOME/coeexpression_oas.jar"
```

- Calls to Owl requires following jar files in the given order:
 - oracleRpasUtils.jar located under \$RPAS_HOME/lib
 - wlfullclient.jar located under \$COE_HOME/java/lib/weblogicJars
 - wseeclient.jar located under \$COE_HOME/java/lib/weblogicJars/wsee
 - ant-all.jar located under \$COE_HOME/java/lib/weblogicJars/ant1.6.5
 - coeexpression_owl.jar located under \$COE_HOME/java/lib

This can be set on Unix as follows:

```
export
RPAS_JAVA_CLASSPATH="$RPAS_HOME/lib/oracleRpasUtils.jar:$COE_
HOME/java/lib/weblogicJars/wlfullclient.jar:$COE_
HOME/java/lib/weblogicJars/wsee/wseeclient.jar:$COE_
HOME/java/lib/weblogicJars/ant1.6.5/ant-all.jar:$COE_
HOME/java/lib/coeexpression_owl.jar"
```

Structure of the Special Expression

The COE expression requires arguments to hold the information necessary to establish the connection with the COE service, provide the desired results, and return the results of the COE service call back to IP/RPAS.

The COE expression makes use of labeled arguments. As a result, the ordering of the arguments is not fixed. However, each argument must be associated with the appropriate label in the following format:

Label:Argument, Label:Argument, ...

The input and output arguments are described in [Table 8–8](#) and [Table 8–9](#). Many of these arguments are either constant values or scalar measure references. Some are dimensional measure references. For dimensional measure references, additional care must be taken to ensure that the measure used conforms to an appropriate intersection.

Input Arguments Table 8-8 describes the input arguments for the COE special expression. Note the following about BaseIntx:

- OptIntx for the dimensional measures is the intersection over which optimization is being performed, as defined by the COE implementation that is acting as the data source for RPAS for this domain.
- BaseIntx is identical to OptIntx, with the addition of a dimension from the Calendar (CLND) hierarchy.
- OptClnD is the dimension from the CLND hierarchy that has been added to the OptIntx to form the BaseIntx.

Table 8-8 COE Special Expression Input Arguments

Name	Type	BaseIntx	COE Semantics	Description
cntry	String	Scalar	forecastRequest.locale.country	Name of the country.
curinv	Double	Opt Intx	forecastRequest.scenario.currentInventory	Current inventory units. Must be greater than or equal to 1.
flgtyp	String	Scalar	forecastRequest.outputFlags.flagType	One of the following optimization types: - forecast - forecast and scenario - forecast and markdown - forecast, markdown, and scenario
ladild	Int	Base Intx	forecastRequest.scenarioMkdn.ladderId	Ladder ID. This is the selected price ladder in the markdown week.
ladval	Double	Base Intx	forecastRequest.scenarioMkdn.ladderValue	Ladder value. Depending on the What-If price type, one of the following values: - What-If Clearance Retail Price at Week -What-If Clearance Retail Price % off at Week
lang	String	Scalar	forecastRequest.locale.language	Name of the language used.
logmsg	Boolean	Scalar	forecastRequest.logOptMsg	Flag to enable the logging of optimization messages (both input and output XML objects) for the current What-If scenario.
mask	Boolean	Opt Intx	N/A. This is used by RPAS to determine the product/location combination to be passed to the COE Service.	Determines the combination of product and location passed in the call to the COE Service.
outDate	Date	Opt Intx	forecastRequest.scenario.outDate	Internal out date.
salval	Double	Opt Intx	forecastRequest.scenario.salvageValue	Salvage value percent off.
selthrpct	Double	Opt Intx	forecastRequest.scenario.sellThruPct	Sell thru percent.
svcurl	String	Scalar	N/A. This is used by RPAS to retrieve the COE Service URL string.	Specifies the measure that holds the COE Service URL.
ver	String	Scalar	forecastRequest.locale.version	Version string.

Output Arguments Table 8–9 describes the output arguments for the COE special expression.

Table 8–9 COE Special Expression Output Arguments

Name	Type	BaseIntx	COE Semantics	Description
desc	String	Opt Intx	forecastResponse.operationStatus.description	Service call description.
err	String	Opt Intx	COEServiceError, exception returned by the COE Service.	Error message returned by the COE Service.
frcst	Int	Base Intx	forecastResponse.forecasts.recForecast.forecastSales	Revised forecasted sales.
itmcurinv	Double	Opt Intx	forecastResponse.forecasts.scenario.currentInventory	Revised current inventory units.
itmoutDate	Date	Opt Intx	forecastResponse.forecasts.scenario.outDate	Revised internal out date.
itmsalval	Double	Opt Intx	forecastResponse.forecasts.scenario.salvageValue	Revised end-of-life (EOL) salvage value percent off.
itemselthrpct	Double	Opt Intx	forecastResponse.forecasts.scenario.sellThruPct	Revised EOL sell thru percent.
prctype	String	Base Intx	forecastResponse.forecasts.mkdnRec.priceType	Revised price type.
recprc	Double	Base Intx	forecastResponse.forecasts.mkdnRec.recPrice	Revised recommended price percent off original price.
saleprc	Double	Base Intx	forecastResponse.forecasts.recForecast.salesPrice	Revised sales average unit retail (AUR).
statcod	Opt Intx	Scalar	forecastResponse.operationStatus.statusCode	Service call status code.
stattyp	Opt Intx	Scalar	forecastResponse.operationStatus.statusType	Service call status type.
tickprc	Double	Base Intx	forecastResponse.forecasts.recForecast.ticketPrice	Revised clearance markdown retail price plan.

Batch Structure Overview

The following directories are used by the batch scripts. These directories are subdirectories of the `$COE_HOME` directory. The extracted files are pipe-delimited flat files.

Table 8–10 Directories Used by Batch Scripts

Directory Name	Content of the Directory
bin	Batch scripts
config	IP template configuration
domain	Domains
input	Input files for building the domain
logs	Log files from running any of the batch scripts A system administrator can scan the logs for any errors, exceptions, or failures. If there are none, the batch completed successfully.
temp	Temporary files used by the batch scripts

Transformation Scripts

There are two scripts used for transforming data across COE and IP/RPAS. Both scripts reside in the `$COE_HOME/bin` directory. There are no arguments for the scripts.

- `coe2rpas.sh` is used for transforming COE data into IP-COE data. The COE files that are present in the `$COE_HOME/bin/fromCOE` directory are transformed and put into the input directory of the IP-COE domain.
- `rpas2coe.sh` is used for exporting and transforming IP-COE data into COE compatible data. The IP-COE files are exported into the `$COE_HOME/bin/toCOE` directory and transformed into the output directory of the IP-COE domain.

Batch Scheduling

Scheduling of the batch scripts are in the following categories:

- [Daily Batch Scripts](#)
- [Weekly Batch Scripts](#)
- [Unscheduled Administration Script](#)

The following information is included in the tables for each batch script:

- A short description of the script
- The name of the script
- The subdirectory in the `$COE_HOME/bin` directory where the batch script resides
- Dependencies on other batch scripts

Daily Batch Scripts

These scripts are run every day. They are run before executing the weekly batch scripts.

Table 8–11 lists information on the daily batch scripts.

Table 8–11 Daily Batch Scripts

Description	Script Name	Batch Directory	Dependency
Backup	N/A	N/A	N/A
Export data to MFP	exportToMFP.ksh	bin	Backup
Load Actuals	loadActuals.ksh	bin	Backup

Weekly Batch Scripts

The daily batch scripts are run before executing the weekly batch scripts.

Table 8–12 lists information on the weekly batch scripts.

Table 8–12 Weekly Batch Scripts

Description	Script Name	Batch Directory	Dependency
Backup	N/A	N/A	N/A
Export data to MFP	exportToMFP.ksh	bin	Backup
Export and Transformation of COE Initial Parameters	rpas2coe.sh	bin	Backup
Formalize DPM Positions	updateDpmPositionStatus.ksh	loadhier	Backup
Calendar Hierarchy Load	loadhier.ksh	loadhier	Backup
Product Hierarchy Load	loadhier.ksh	loadhier	Backup, Formalize DPM Positions (if run)
Location Hierarchy Load	loadhier.ksh	loadhier	Backup
Load on order data	loadActuals.ksh	bin	Calendar Hierarchy Load, Product Hierarchy Load, Location Hierarchy Load
Load MFP data	loadActuals.ksh	bin	Calendar hierarchy load, Product hierarchy load, Location hierarchy load
Load Actuals data	loadActuals.ksh	bin	Calendar hierarchy load, Product hierarchy load, Location hierarchy load
Load AP data	loadActuals.ksh	bin	Calendar hierarchy load, Product hierarchy load, Location hierarchy load
Transform COE Batch Files	coe2rpas.sh	bin	Backup

Table 8–12 (Cont.) Weekly Batch Scripts

Description	Script Name	Batch Directory	Dependency
Load COE Batch	loadActuals.ksh	bin	Calendar hierarchy load, Product hierarchy load, Location hierarchy load
Propagate inventory and aggregate data for all planning levels	processactuals.ksh	actualize	Load on order data, Load MFP data
Generate sales forecast	runforecast.ksh	forecast	Load on order data, Load MFP data
Refresh existing workbooks Note: This script is optional.	refresh.ksh	workbook	runforecast
Auto build workbooks placed on queue Note: This script is optional.	autobuild.ksh	workbook	None

Unscheduled Administration Script

The following script is not part of a normal batch schedule. This script is run only to perform the specified activity.

[Table 8–13](#) lists information on the unscheduled administration script. This script is located in the \$COE_HOME/bin directory.

Table 8–13 Unscheduled Administration Script

Description	Script Name	Batch Directory	Dependency
Load mapping	loadActuals.ksh	bin	None

Batch Environment Scripts

These scripts are included in the other batch scripts to control logging and set environment variables. These batch scripts are only supported for IP Configured for COE.

The first script, `message.ksh`, controls the overall logging. The script writes batch script details to a daily log file. The daily log file is created in the `$COE_HOME/logs` directory. The format of the file name is `MnthID_Day.log`, for example, `Apr_02.log`.

The second script, `environment.ksh`, is called at the beginning of every batch script. This script sets the following environment variables:

- `export ITEM_CONFIGNAME=itemplan`
- `export ITEM_DOMAINHOME=$COE_HOME/domain`
- `export ITEM_MASTERDOMAIN=$ITEM_DOMAINHOME/itemplan`
- `export ITEM_CONFIGHOME=$COE_HOME/config`
- `export ITEM_EXPORT=$COE_HOME/export`
- `export ITEM_INPUThOME=$COE_HOME/input`
- `export ITEM_LOG_DIR=$COE_HOME/logs`
- `export ITEM_LIB=$COE_HOME/bin`
- `export ITEM_TEMP=$COE_HOME/temp`
- `export ITEM_BATCH=$COE_HOME/bin`
- `export LOGLEVEL=all`
- `export RECORDLOGLEVEL=warning`

The `LOGLEVEL` and `RECORDLOGLEVEL` parameters can be set to any of the RPAS supported logging levels—all, profile, debug, audit, information, warning, error, and none.

Batch Designs

This section contains detailed information on the following batch script:

- [Load COE Batch](#)

For information on the following scripts, see [Transformation Scripts](#):

- Export and Transformation of COE Initial Parameters
- Transform COE Batch Files

For a detailed description of the other scripts, see [Batch Designs](#) in [Chapter 7](#).

Load COE Batch

Script

`loadActuals.ksh`

Usage

`loadActuals.ksh <measurelist> <maxprocesses>`

Argument	Description	Notes
measurelist	Sets the location of the file which contains the list of measures to be exported.	By default, these files are provided with the package in the \$COE_HOME/bin directory.
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.

Control File

LoadCOEBatch.txt: Contains the following list of measures that can be loaded as part of this script:

- ipwpcorcmdppv
- ipwpcopeopiuv
- ipwpcorcfcsuv
- ipwpcorcopcov
- ipwpcoprslsarv
- coewiprcldid

Example

loadActuals.ksh LoadCOEBatch.txt 1

Error Information

Table 8–14 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	One or more arguments are missing.
loadactuals	41	yes	Domain does not exist.
loadactuals	42	no	Data file does not exist.
loadactuals	43	yes	All measure input files are empty or missing.
loadactuals	44	yes	Configuration file does not exist.
loadactuals	45	yes	Errors occurred during the load of one or more measures.

Notes

- This script uses the RPAS loadmeasure utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The script ignores any missing or empty measure load files.
- All measure files are placed into the domain's input folder.
- If there were no errors during the loading of all measures, the input file is archived into the domain's input/processed directory. A date stamp is appended to the end of the measure file name.
- The script does not produce an error when records are rejected from the loaded files. These rejected records are logged in the log output of the load process.

