Oracle® Retail Item Planning

Operations Guide

Release 14.0.2

E60312-01

January 2015

ORACLE®

Oracle® Retail Item Planning Operations Guide, Release 14.0.2

E60312-01

Primary Author: Nirmala Suryaprakasha

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

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

Contents

3 Build Scripts

4 Data Flow

5 Fashion Planning and ODI Integration

6 Script Integration

7 Batch Processing

8 Item Planning Configured for COE

List of Tables

List of Figures

Send Us Your Comments

Oracle Retail Item Planning Operations Guide, Release 14.0.2

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

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

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

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

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network web site. It contains the most current Documentation Library plus all documents revised or released recently.

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

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

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

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

Preface

This Operations Guide provides critical information about the processing and operating details of Oracle Retail Item Planning, including the following:

- System configuration settings
- Technical architecture
- Functional integration dataflow across the enterprise
- Batch processing

Audience

This guide is for:

- Systems administration and operations personnel
- Systems analysts
- Integrators and implementers
- Business analysts who need information about Oracle Retail Item Planning processes and interfaces

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Retail Item Planning Release 14.0.2 documentation set:

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

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

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

For more information about the RPAS and Fusion Client, see the documents in the Oracle Retail Predictive Application Server documentation set.

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

- Oracle Retail Predictive Application Server documentation
- Oracle Data Integrator documentation
- Oracle Retail ODI Enabled Integration 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, 14.0) or a later patch release (for example, 14.0.1). If you are installing the base release or additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

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/technetwork/documentation/oracle-retail-100266.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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction

Oracle Retail 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 Item Planning, see the *Oracle Retail Item Planning User Guide for the Retail Predictive Application Server Classic Client* or *Oracle Retail Item Planning User Guide for the Retail Predictive Application Server Fusion Client*.

Contents of this Guide

This operations guide addresses the following topics:

- Chapter 1, "Introduction"—Overview of the Item Planning 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 Item Planning domain.
- Chapter 4, "Data Flow"—Overview of the Item Planning data flow.
- Chapter 5, "Fashion Planning 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.
- Chapter 9, "Internationalization"—Translations provided for SPO.

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

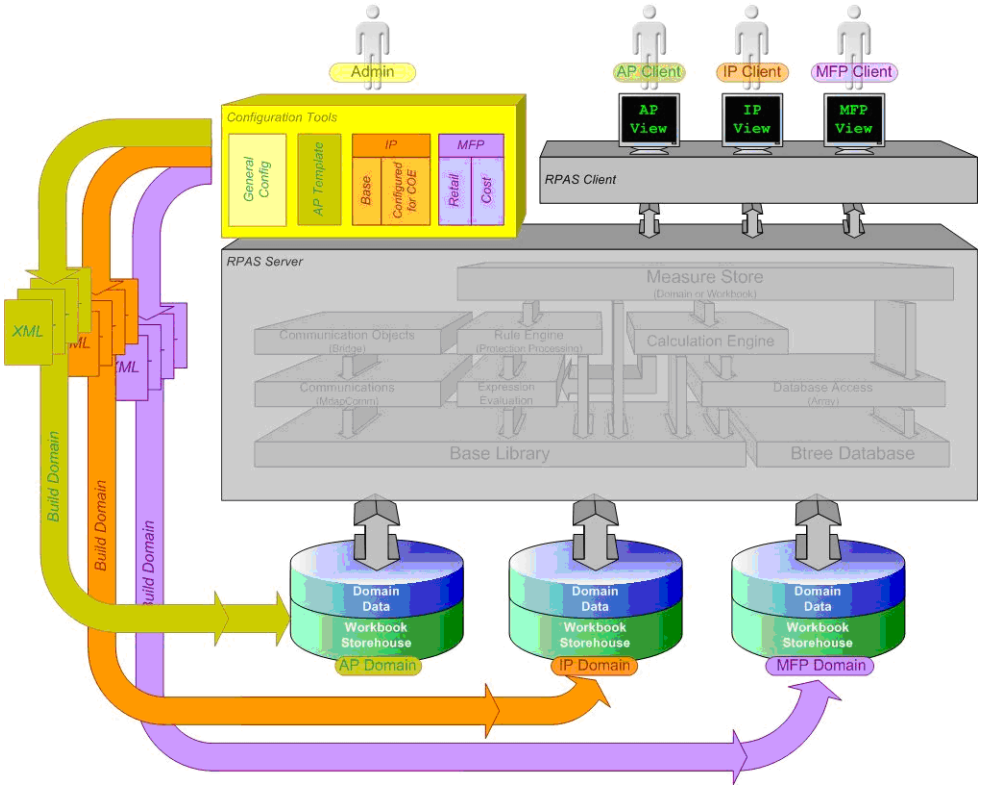
Planning (MFP), and Size Profile Optimization (SPO) as a full-suite planning solution for fashion retailers.

For details on the integration, see Chapter 5, "Fashion Planning and ODI Integration" and Chapter 6, "Script Integration".

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

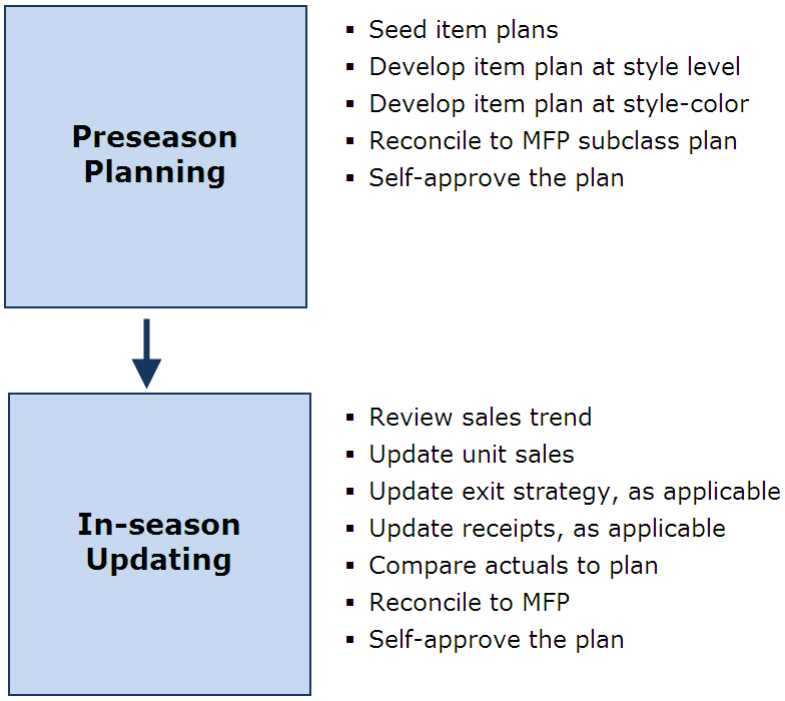
A client requests to use one of the templates via the Configuration Tools. A number of XML files are then output which define the domain to be created. These XML files are used to build the specified domain incorporating all the attributes mentioned above that have been defined specifically.

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 Item Planning.

Figure 1–2 Business Process Workflow



Key Features of Item Planning

Item Planning provides the following features:

- 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 (RPAS)
- Oracle Retail Assortment Planning (AP)
- Oracle Retail Item Planning (IP)
- Oracle Retail Clearance Optimization Engine (COE)
- Oracle Retail Merchandise Financial Planning (MFP)
- Oracle Retail Size Profile Optimization (SPO)
- Oracle Retail Data Warehouse (RDW)
- Oracle Retail Merchandising System (RMS)

Note: RPAS stores hierarchy position names in lower case. The hierarchy load files in RPAS for IP-COE can be in mixed case; however, they are stored in lower case internally. The conversion on the IP side is handled by the convertDomain utility, and no manual loads need to be performed.

COE, on the other hand, supports mixed case position names (that is, merchandise_key and location_key), both in load files and in database storage. Moreover, COE uses case-sensitive comparisons for position names (ids). As a result, you must load lower case position names (that is, merchandise_key and location_key) in COE.

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 Item Planning
- Retailer's hierarchical (SKU/store/day) data
- Item Planning 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

2

Implementation Considerations

The following information needs to be considered before configuring Item Planning:

- Historical Data
- Hardware Space 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 Item Planning. With less than two years of history, the forecast may not generate the optimal results.

Hardware Space Impacts

The following factors can affect hardware space 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 Item Planning 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 Item Planning 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 Item Planning 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 for the Classic Client* or *Oracle Retail Predictive Application Server User Guide for the Fusion Client*.

Plug-ins

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

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 Item Planning 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.

Note: Starting with Release 13.3, RPAS uses integer indexing for hierarchy positions. As a result, the upgrade process of a given solution includes a conversion process. For more information on the migration process to RPAS 13.3, refer to the *Oracle Retail Predictive Application Server Administration Guide*.

Complete the steps in the *Oracle Retail Predictive Application Server Administration Guide* for converting and upgrading your domain.

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.

- 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 you save a workbook, you can 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 for the Classic Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Fusion Client*.

Internationalization

Internationalization is the process of creating software that can be translated more easily. Changes to the code are not specific to any particular market.

Oracle Retail applications have been internationalized to support multiple languages.

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 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.

3

Build Scripts

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

Installation Dependencies

RPAS and Item Planning must be installed before setting up and configuring Item Planning.

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

export IP_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

RPAS Client Installation

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

- RPAS Classic Client—A Windows-based client interface for end users and system administrators of an RPAS domain.

- RPAS Fusion Client—A web-based client developed using Oracle Application Development Framework (ADF).

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

Item Planning Installation

The Item Planning installer performs the following functions:

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

Custom Domain Build

To do a custom build of a domain:

1. Change to the configuration directory: cd $IP_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 $IP_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

   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.

Item Planning Taskflow for the RPAS Fusion Client

The Item Planning installation software enables you to install the taskflow and online help files for the RPAS Fusion Client. In order to install the taskflow files, the RPAS Fusion Client must already be installed. For more information on installing the RPAS Fusion Client, see the *Oracle Retail Predictive Application Server Installation Guide*.

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

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 Item Planning 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 the Fusion Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Classic Client* for details on the RPAS utilities.

Files Needed to Build the IP Domain

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

- Standard RPAS Hierarchy Files
- IP Specific Hierarchy Files

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

Standard RPAS Hierarchy Files

The following hierarchy files are needed:

- Calendar Hierarchy File
- Product Hierarchy File
- Location Hierarchy File

Calendar Hierarchy File

File name: clnd.hdr.csv.dat

File format: comma-separated value (CSV) file

The following table describes the fields in this file.

Table 3–1 Calendar Hierarchy File

Name	Label	Description	Hierarchy Type	Child
day	Day	Day or Date in YYYYMMDD format	Main	None
week	Week	Week Number	Main	day
mnth	Month	Month Number	Main	week
qrtr	Quarter	Quarter of the Year	Main	mnth
ssn	Half	Half Year	Main	qrtr
year	Year	Year	Main	ssn
woy	Week of Year	Week of Year	Alternate	week

Example:

day, day_label, week, week_label,mnth, mnth_label, qrtr, qrtr_label, ssn, ssn_label, year, year_label, woy, woy_label

20070121,1/21/2007,W01_2007,1/27/2007,M01_2007,Feb FY2007,Q1_2007,Q1 FY2007,H1_2007,Half1 FY2007,A2007,FY2007,01,Week 01

20070122,1/22/2007,W01_2007,1/27/2007,M01_2007,Feb FY2007,Q1_2007,Q1 FY2007,H1_2007,Half1 FY2007,A2007,FY2007,01,Week 01

20070123,1/23/2007,W01_2007,1/27/2007,M01_2007,Feb FY2007,Q1_2007,Q1 FY2007,H1_2007,Half1 FY2007,A2007,FY2007,01,Week 01

20070124,1/24/2007,W01_2007,1/27/2007,M01_2007,Feb FY2007,Q1_2007,Q1 FY2007,H1_2007,Half1 FY2007,A2007,FY2007,01,Week 01

Product Hierarchy File

File name: prod.hdr.csv.dat

File format: comma-separated value (CSV) file

The following table describes the fields in this file:

Table 3–2 Product Hierarchy File

Name	Label	Description	Hierarchy Type	Child
skup	Style-Color	Style-Color	Main	None
skug	Style	Style	Main	skup
scls	Subclass	Subclass	Main	skug
clss	Class	Class	Main	scls
dept	Dept	Department	Main	clss
pgrp	Group	Group	Main	dept
dvsn	Division	Division	Main	pgrp
cmpp	Company	Company	Main	dvsn
clr	Color	Color	Alternate	skup

Example:

sku,sku_label,skup,skup_label,skug,skug_label,scls,scls_label,clss,clss_label,dept,dept_label,pgrp,pgrp_label,dvsn,dvsn_label,cmpp,cmpp_label,clr,clr_label

1000000_6,1000000_6 Suede Loafer Brown size 6,1000000,1000000 Suede Loafer Brown,200000,200000 Suede Loafer,30000,30000 Loafers,4000,4000 Casual,100,100 Men's Footwear,21,21 Men's Footwear,20,20 Menswear,1,1 Acme Home,BROWN,Brown

1000001_6,1000001_6 Suede Loafer Black size 6,1000001,1000001 Suede Loafer Black,200000,200000 Suede Loafer,30000,30000 Loafers,4000,4000 Casual,100,100 Men's Footwear,21,21 Men's Footwear,20,20 Menswear,1,1 Acme Home,BLACK,Black

Location Hierarchy File

File name: loc.hdr.csv.dat

File format: comma-separated value (CSV) file

The following table describes the fields in this file:

Table 3–3 Location Hierarchy File

Name	Label	Description	Hierarchy Type	Child
str	Store	Store	Main	None
pzon	Price Zone	Price Zone	Main	str
zone	Zone	Zone	Main	pzon
chnl	Channel	Channel	Main	zon
chn	Chain	Chain	Main	chnl
cmpy	Company	Company	Main	chn

Example:

1000,1000 Charlotte,PZ1,FB PriceZone 1,Z1,FB Zone 1,1,Brick & Mortar,1,Chain 1,1,Retailer Ltd

1001,1001 Atlanta,Unassigned,No price zone,Unassigned,No zone,1,Brick & Mortar,1,Chain 1,1,Retailer Ltd

IP Specific Hierarchy Files

The following are special IP Hierarchy files.

- Seasonality Hierarchy File
- Price Tier Hierarchy File
- Size Hierarchy File
- Curve Library Hierarchy File
- Measure_Status Hierarchy File
- Measure Details Hierarchy File
- Promotion Hierarchy File
- Clearance Cadence Hierarchy File
- Week of Year Hierarchy File

Seasonality Hierarchy File

This hierarchy is a single dimension hierarchy that contains the seasonality type of items.

File name: at1h.csv.dat

File format: comma-separated value (CSV) file

The following table describes the fields in this file:

Table 3–4 Seasonal Hierarchy File

Name	Label	Description	Hierarchy Type	Child
at1d	Seasonality	Seasonality	Main	None

Example:

01,Fashion

05,Seasonal Basic

99,Basic

Price Tier Hierarchy File

This hierarchy is a single dimension hierarchy that contains the price tiers.

File name: at2h.csv.dat

File format: comma-separated value (CSV) file

The following table describes the fields in this file.

Table 3–5 Price Tier Hierarchy File

Name	Label	Description	Hierarchy Type	Child
at2d	Price Tier	Price Tier	Main	None

Example:

1,1 Good

2,2 Better

3,3 Best

Size Hierarchy File

The Size hierarchy captures all valid size ranges and corresponding sizes in each size range.

File name: sizh.csv.dat

File format: comma-separated value (CSV) file

The following table describes the fields in this file:

Table 3–6 Size Hierarchy File

Name	Label	Description	Hierarchy Type	Child
sizd	Size	Size identifier. Note: The size is expected to have a unique identifier. For example an S in an S-M-L size range needs to have a unique identifier from an S in an S-M-L-XL size range	Main	None
srng	Size Range	Unique size range identifier. The size range is a hierarchical aggregate of size.	Main	Sizd

Example:

XS_SML,XS,SML,SML

S_SML,S,SML,SML

M_SML,M,SML,SML

L_SML,L,SML,SML

XL_SML,XL,SML,SML

XXL_SML,XXL,SML,SML

04_0t15,4,0t15,0-15

06_0t15,6,0t15,0-15

08_0t15,8,0t15,0-15

10_0t15,10,0t15,0-15

12_0t15,12,0t15,0-15

14_0t15,14,0t15,0-15

16_0t15,16,0t15,0-15

18_0t15,18,0t15,0-15

6_MensShoes,6,MensShoes,Men's Shoes

6_5_MensShoes,6.5,MensShoes,Men's Shoes

Curve Library Hierarchy File

This hierarchy is a single dimension hierarchy that contains the positions of different sales curves.

File name: libh.csv.dat

File format: comma-separated value (CSV) file

The following table describes the fields in this file:

Table 3–7 Curve Library Hierarchy File

Name	Label	Description	Hierarchy Type	Child
libd	Curve Library	Curve Library	Main	None

Example:

01,Lifecycle Curve 01

02,Lifecycle Curve 02

03,Lifecycle Curve 03

04,Lifecycle Curve 04

05,Lifecycle Curve 05

06,Lifecycle Curve 06

07,Lifecycle Curve 07

08,Lifecycle Curve 08

Measure_Status Hierarchy File

This hierarchy is a single dimension hierarchy that contains the positions of status of measures.

File name: dash.csv.dat

File format: comma-separated value (CSV) file

The following table describes the fields in this file:

Table 3–8 Measure_Status Hierarchy File

Name	Label	Description	Hierarchy Type	Child
dshs	Status	Status	Main	None

Example:

001,01. Total Plan

002,02. Total Act/Proj

003,03. Total Act/Proj diff to Total Plan

004,04. Total Act/Proj var to Total Plan

005,05. LTD Plan

006,06. LTD Actual

007,07. LTD Actuals diff to Plan

008,08. LTD Actuals var to Plan

009,09. Remaining Plan

010,10. Remaining Projection

011,11. Remaining Projection diff to Plan

012,12. Remaining Projection var to Plan

013,13. LTD Plan % of Plan

014,14. LTD Actual % of Plan

Measure Details Hierarchy File

This hierarchy is a single dimension hierarchy that contains the positions of details of measures.

File name: dsh2.csv.dat

File format: comma-separated value (CSV) file

The following table describes the fields in this file:

Table 3–9 Measure Details Hierarchy File

Name	Label	Description	Hierarchy Type	Child
detl	Measure Detail	Measure Detail	Main	None

Example:

01,01. Total Plan

02,02. Total Act/Proj

03,03. Total Act/Plan diff to Total Plan

04,04. Total Act/Proj var to Total Plan

05,05. Total Act/Initial Optimization Forecast

06,06. Total Act/Initial Opt Fcst diff to Total Plan

07,07. Total Act/Initial Opt Fcst var to Total Plan

08,08. Total Act/Revised Optimization Forecast

09,09. Total Act/Rev Opt Fcst diff to Total Plan

10,10. Total Act/Rev Opt Fcst var to Total Plan

Promotion Hierarchy File

This hierarchy is a single dimension hierarchy that contains the positions of promotions.

File name: prom.csv.dat

File format: comma-separated value (CSV) file

The following table describes the fields in this file:

Table 3–10 Promotion Hierarchy File

Name	Label	Description	Hierarchy Type	Child
pro	Promotion	Promotion	Main	None

Example:

Prom01,Promotion 01

Prom02,Promotion 02

Prom03,Promotion 03

Prom04,Promotion 04

Prom05,Promotion 05

Prom06,Promotion 06

Prom07,Promotion 07

Prom08,Promotion 08

Prom09,Promotion 09

Prom10,Promotion 10

Prom11,Promotion 11

Prom12,Promotion 12

Clearance Cadence Hierarchy File

This hierarchy is a single dimension hierarchy that contains the positions of Clearance Cadence.

File name: clsh.csv.dat

File format: comma-separated value (CSV) file

The following table describes the field in this file:

Table 3–11 Clearance Cadence Hierarchy File

Name	Label	Description	Hierarchy Type	Child
clrc	Clearance Cadence	Clearance Cadence	Main	None

Example:

01,Cadence 01

02,Cadence 02

03,Cadence 03

04,Cadence 04

05,Cadence 05

Week of Year Hierarchy File

This hierarchy is a single dimension hierarchy that contains the positions of week of year.

File name: woy2.csv.dat

File format: comma-separated value (CSV) file

The following table describes the field in this file:

Table 3–12 Week of Year Hierarchy File

Name	Label	Description	Hierarchy Type	Child
wy2	Week of Year	Week of Year	Main	None

Example:

01,Week 01

02,Week 02

03,Week 03

04,Week 04

05,Week 05

06,Week 06

Building the Item Planning Domain

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

Table 3–13 outlines the build scripts. These scripts are located in the `$IP_HOME/bin` directory.

Table 3–13 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 3–14 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.

Table 3–14 (Cont.) Building a Domain Error Information

Task Name	Error Code	Abort Required?	Description of Error
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 the Fusion Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Classic Client* 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 the Fusion Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Classic Client* for details on these utilities.
- All hierarchy and measure files are placed into the `$IP_HOME/input` directory.
- The commit later functionality and insert measure functionality are disabled.

Patching a Domain

When changes have been made to Item Planning 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 3–15 Patching a Domain Error Information

Task Name	Error Code	Abort Required?	Description of Error
patch_item_deleteformats	-10	yes	Could not find the localdomainlist.cfg file.
patch_item_deleteformats	-11	yes	Error during domain patch.
patch_item_keepformats	-20	yes	Error during domain patch.

Notes

- Prior to RPAS 13.1.1, formats were not transferable 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 the Fusion Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Classic Client* 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 for the Fusion Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Classic Client*.

Loading and Extracting Data

Data is loaded into Item Planning using the standard RPAS approach. See the *Oracle Retail Predictive Application Server Administration Guide for the Fusion Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Classic Client* 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, "Fashion Planning and ODI Integration". If you are using script integration for the bundle, see Chapter 6, "Script Integration". For information on any other batch script, see Chapter 7, "Batch Processing".

4

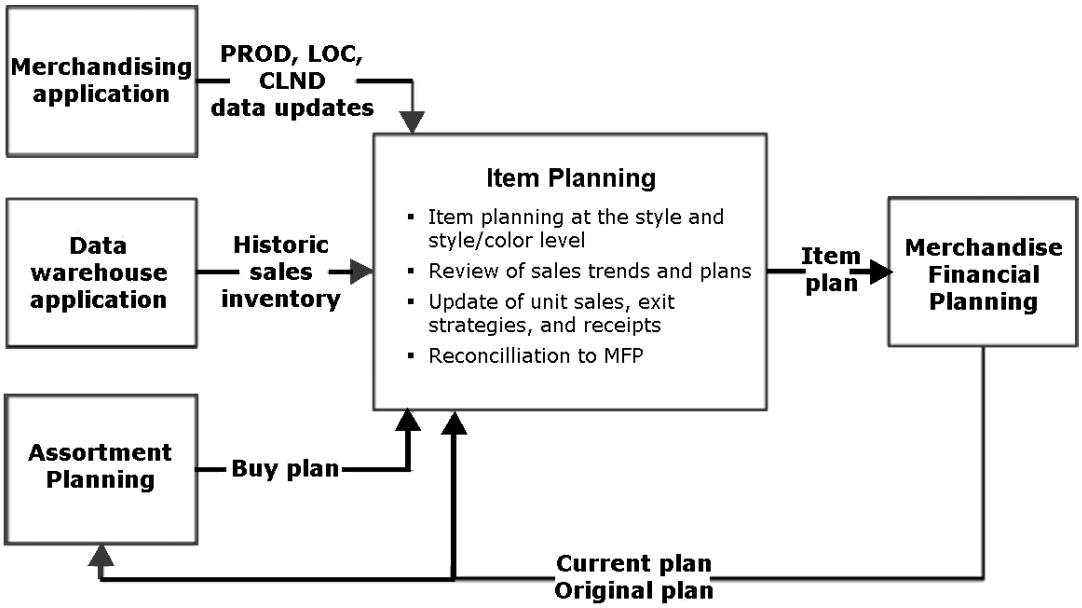
Data Flow

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

Overview of the Item Planning Data Flow

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

Figure 4–1 Data Flow for the Integration of Item Planning 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, "Fashion Planning and ODI Integration". For information on the interface between IP and COE, see Chapter 8, "Item Planning Configured for COE".

From a Merchandising Application to Item Planning

Data for the following hierarchies is imported into Item Planning 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 Item Planning from a data warehouse application:

- Historic sales
- Inventory

5

Fashion Planning 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, "Script Integration".

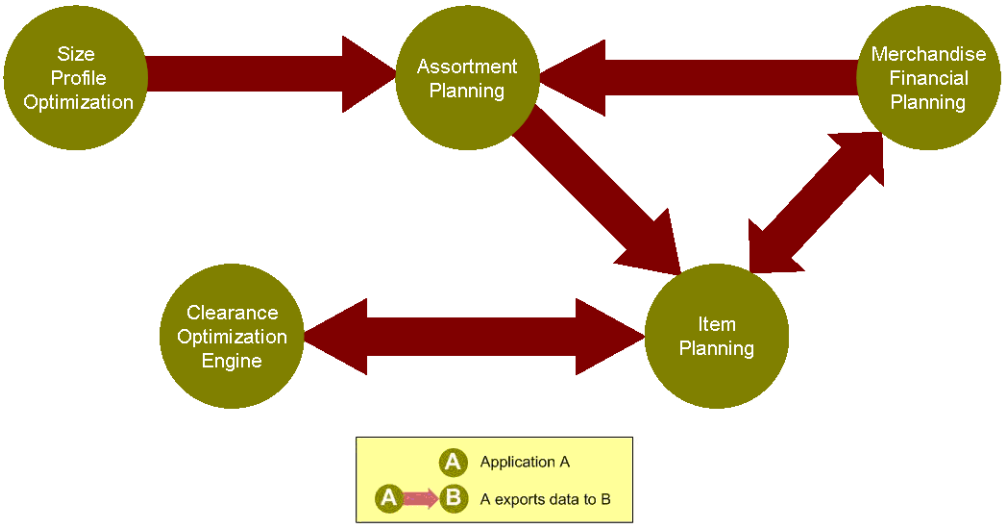
Overview of the Fashion Planning Bundle

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 applications: Item Planning (IP), Item Planning Configured for Clearance Optimization Engine (IP 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 the 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
- Prepack definition configuration at style-color/size/prepack-ID and prepack validity periods at prepack-ID week

The size profiles are used by AP to obtain more detailed buy plans at style-color/size or prepack per store per receipt week. The prepack optimization module of SPO addresses the optimization of not only the number of units within a pack, but also the size ratios for each style-color within the pack. 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 prepacks.

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

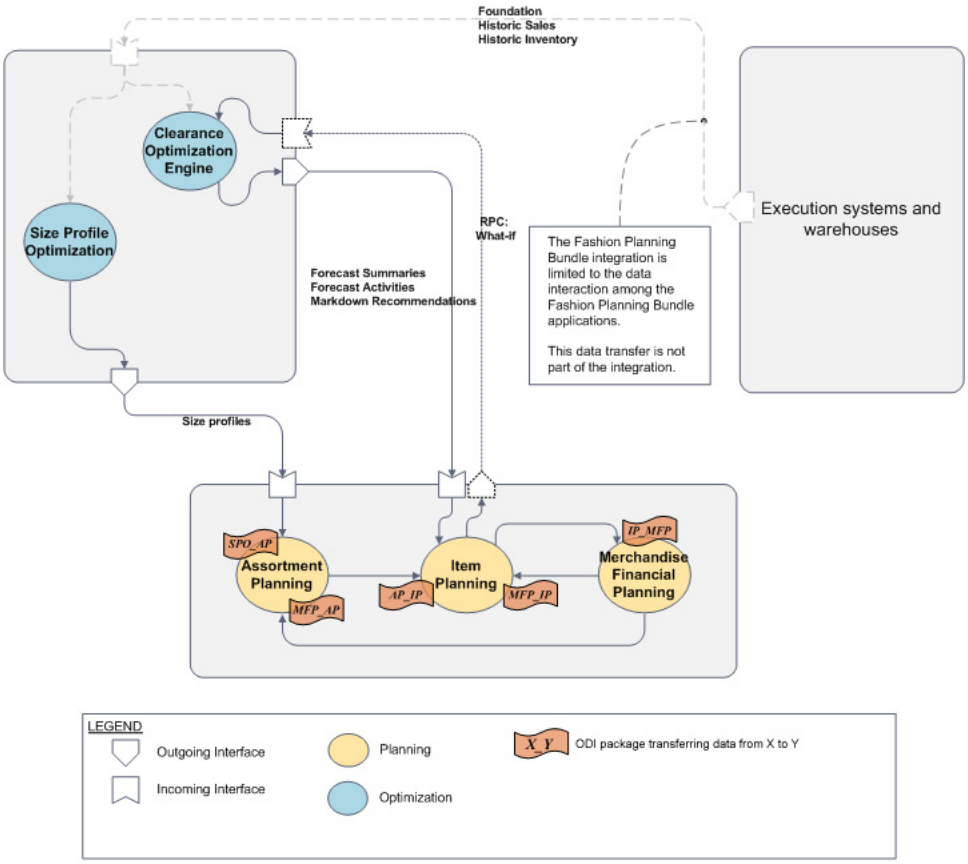
The following data is exported from MFP to AP:

- 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 Integration

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

Oracle Retail Enabled ODI Integration leverages Oracle Data Integrator (ODI) to store information about data interfaces among applications. ODI presents a user-friendly graphical interface for user-initiated data transfers and runtime monitoring. It also provides the ability to host application domains on different machines on a network, an ability not available with prior non-ODI integration strategies.

The Fashion Planning Bundle applications supported by the Oracle Retail Enabled ODI 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 integration, see the *Oracle Retail Enabled Oracle Data Integrator Integration Implementation Guide*.

Measure Data Integration

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

- SizeOpt to AP Package
- SizeOpt to AP with Prepack Package
- IP to MFP Retail Package
- IP to MFP Cost Package
- AP to IP Package
- MFP Cost to AP Package
- MFP Retail to AP Package
- MFP Cost to IP Package
- MFP Retail to IP Package
- MFP Finalize Exports Package

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

$RPAS_HOME/scripts/integration/ODI

SizeOpt to AP Package

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

> **Note:** For customers who use Assortment Planning with the Prepack component, see the SizeOpt to AP with Prepack Package section.

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 the expression is populated with a value other than the default.

Table 5–1 SizeOpt to AP Data

SizeOpt Expression	Expression Type	Expression Default Value	AP Target Measure
eptpackdef	Integer	0	eptpackdef
esprofxxlaxg	Real	-1	sztyarchsp1up
esprofxxlbxg	Real	-1	sztyarchsp2up
esprofxxlcxg	Real	-1	sztyarchsp3up
esprofxxldxg	Real	-1	sztyarachsp4up
sku2sizexxlxg	Boolean	False	sztyclssszmapb
exportprofxxlxg	Real	-1	sztysizeprflup
exportescxxlxg	String	""	szwpescleveltx
sku2atcdxxlxg	Boolean	False	szwpssncodeb

SizeOpt to AP with Prepack Package

The following information is about integrating SizeOpt to AP with Prepack integration.

> **Note:** This package is for customers who use Assortment Planning with the Prepack component.
>
> For customers who use Assortment Planning without the Prepack component, see the SizeOpt to AP Package section.

This package is the same as the SizeOpt to AP Package package, except that the eptpackdef data is not sent from SizeOpt to AP. This allows SizeOpt and AP to have independent versions of the prepack definitions data.

Scripts Used By the SizeOpt to AP with Prepack Package

- sizeopt_to_ap_init.ksh
- sizeopt_to_ap_finalize.ksh

Data Mapping for SizeOpt to AP with Prepack Package

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

Table 5–2 SizeOpt to AP with Prepack Data

SizeOpt Expression	Expression Type	Expression Default Value	AP Target Measure
esprofxxlaxg	Real	-1	sztyarchsp1up
esprofxxlbxg	Real	-1	sztyarchsp2up
esprofxxlcxg	Real	-1	sztyarchsp3up
esprofxxldxg	Real	-1	sztyarachsp4up

Table 5–2 (Cont.) SizeOpt to AP with Prepack Data

SizeOpt Expression	Expression Type	Expression Default Value	AP Target Measure
sku2sizexxlxg	Boolean	False	sztyclssszmapb
sku2sizexxlxg	Boolean	False	skup2msrn
sku2sizexxlxg	Boolean	False	skup2msiz
exportprofxxlxg	Real	-1	sztysizeprflup
exportescxxlxg	String	""	szwpescleveltx
sku2atcdxxlxg	Boolean	False	szwpssncodeb

IP to MFP Retail Package

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

Scripts Used By the IP to MFP Retail Package

- ip_to_mfp_init.ksh
- ip_to_mfp_finalize.ksh

Data Mapping for IP to MFP Retail Package

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

Table 5–3 IP to MFP Retail: CP Approved Data

IP Expression	Expression Type	Expression Default Value	MFP Retail Target Measure
ipcpbopc	Real	0	ipcpbopc
ipcpbopr	Real	0	ipcpbopr
ipcpbopu	Real	0	ipcpbopu
ipcpeopc	Real	0	ipcpeopc
ipcpeopr	Real	0	ipcpeopr
ipcpeopu	Real	0	ipcpeopu
ipcpgmpv	Real	0	ipcpgmpv
ipcprecc	Real	0	ipcprecc
ipcprecr	Real	0	ipcprecr
ipcprecu	Real	0	ipcprecu
ipcpslsr	Real	0	ipcpslsr
ipcpslsu	Real	0	ipcpslsu

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

Table 5–4 IP to MFP Retail: OP Approved Data

IP Expression	Expression Type	Expression Default Value	MFP Retail Target Measure
ipopbopc	Real	0	ipopbopc
ipopbopu	Real	0	ipopbopu

Table 5–4 (Cont.) IP to MFP Retail: OP Approved Data

IP Expression	Expression Type	Expression Default Value	MFP Retail Target Measure
ipopbopr	Real	0	ipopbopr
ipopeopc	Real	0	ipopeopc
ipopeopr	Real	0	ipopeopr
ipopeopu	Real	0	ipopeopu
ipopgmpv	Real	0	ipopgmpv
ipoprecc	Real	0	ipoprecc
ipoprecr	Real	0	ipoprecr
ipoprecu	Real	0	ipoprecu
ipopslsr	Real	0	ipopslsr
ipopslsu	Real	0	ipopslsu

IP to MFP Cost Package

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

Scripts Used By the IP to MFP Cost Package

- ip_to_mfp_init.ksh
- ip_to_mfp_finalize.ksh

Data Mapping for IP to MFP Cost Package

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

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

IP Expression	Expression Type	Expression Default Value	MFP Cost Target Measure
ipcpbopc	Real	0	ipcpbopc
ipcpbopu	Real	0	ipcpbopu
ipcpeopc	Real	0	ipcpeopc
ipcpeopu	Real	0	ipcpeopu
ipcpgmpv	Real	0	ipcpgmpv
ipcprecc	Real	0	ipcprecc
ipcprecu	Real	0	ipcprecu
ipcpslsc	Real	0	ipcpslsc
ipcpslsr	Real	0	ipcpslsr
ipcpslsu	Real	0	ipcpslsu

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

Table 5–6 IP to MFP Cost: OP Approved Data

IP Expression	Expression Type	Expression Default Value	Target MFP Cost Measure
ipopbopc	Real	0	ipopbopc
ipopbopu	Real	0	ipopbopu
ipopeopc	Real	0	ipopeopc
ipopeopu	Real	0	ipopeopu
ipopgmpv	Real	0	ipopgmpv
ipoprecc	Real	0	ipoprecc
ipoprecu	Real	0	ipoprecu
ipopslsc	Real	0	ipopslsc
ipopslsr	Real	0	ipopslsr
ipopslsu	Real	0	ipopslsu

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 the expression is populated with a value other than the default value.

Table 5–7 AP to IP Data

AP Expression	Expression Type	Expression Default Value	IP Target Measure
bpcpasgn2clsb	Boolean	False	apcpasgn2strb
bpcsbopc	Real	0	apcpbopc
bpcsbopr	Real	0	apcpbopr
bpcsbopu	Real	0	apcpbopu
bpcsfpcstu	Real	0	apcpcogsc
bpwcdelfrequ	Integer	1	apcpdelfrequ
bpcseopc	Real	0	apcpeopc
bpcseopr	Real	0	apcpeopr
bpcseopu	Real	0	apcpeopu
bpcpexitweekdt	Date	[Jan 1, 1900]	apcpexitd
acwpfabrictx	String	""	apcpfabrictx
sptyitmat1maptx	String	""	apcpitmat1maptx
sptyitmat2maptx	String	""	apcpitmat2maptx
bpcsmkdclrr	Real	0	apcpmkdclrr

Table 5–7 (Cont.) AP to IP Data

AP Expression	Expression Type	Expression Default Value	IP Target Measure
bpcsmkdpermr	Real	0	apcpmkdpermr
bpcsmkdpromor	Real	0	apcpmkdpror
bpcpprepacku	Integer	1	apcppckszu
sccpperfgrpll	String	""	apcpperfgrptx
bpcsclrcstu	Real	0	apcpprcclrc
bpcsclrprcu	Real	0	apcpprcclrr
bpcsfprtlu	Real	0	apcpprcinir
bpcspromoprcr	Real	0	apcpprcpror
bpcspermprcr	Real	0	apcpprcr
bpwcpresminu	Integer	0	apcppresminu
bpcsrecc	Real	0	apcprecc
bpcsrecr	Real	0	apcprecr
bpcsrecu	Real	0	apcprecu
sccpsizgrpll	String	""	apcpsftystcku
acwpsizrangetx	String	""	apcpsizgrptx
actyvendorll	String	""	apcpsizrangetx
bpcsslsclrc	Real	0	apcpslsclrc
bpcsslsclrr	Real	0	apcpslsclrr
bpcsslsclru	Real	0	apcpslsclru
bpcsfpslsc	Real	0	apcpslsregc
bpcsfpslsr	Real	0	apcpslsregr
bpcsfpslsu	Real	0	apcpslsregu
bpcpstartweekdt	Date	[Jan 1, 1990]	apcpslsstartd
bpcsstrcntu	Real	0	apcpstru
actyvendorll	String	""	apcpvendortx
bpcpwksclru	Integer	0	apcpweeksclru
bpcpwksregu	Integer	0	apcpweeksregu

MFP Cost to AP Package

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

Scripts Used By the MFP Cost to AP Package

- mfp_to_ap_init.ksh
- mfp_to_ap_finalize.ksh

Data Mapping for MFP Cost to AP Package

Data is sent when mowpappcpnewb is set to True.

Table 5–8 MFP Cost to AP Data

MFP Cost Expression	Expression Type	Expression Default Value	AP Target Measure
bucpgmpv	Real	0	mfcpposgmr
bucprecc	Real	0	mfcprecc
bucprecu	Real	0	mfcprecu
bucpslsr	Real	0	mfcpslsr
bucpslsu	Real	0	mfcpslsu

MFP Retail to AP Package

The following information is about the MFP Retail to AP integration.

Scripts Used By the MFP Retail to AP Package

- mfp_to_ap_init.ksh
- mfp_to_ap_finalize.ksh

Data Mapping for MFP Retail to AP Package

Data is sent when mowpappcpnewb is set to True.

Table 5–9 MFP Retail to AP Data

MFP Retail Expression	Expression Type	Expression Default Value	AP Target Measure
bucpagmpv	Real	0	mfcpposgmr
bucprecc	Real	0	mfcprecc
bucprecr	Real	0	mfcprecr
bucprecu	Real	0	mfcprecu
bucpslsregr + bucpslsclrr + bucpslspror	Real	0	mfcpslsr
bucpslsregu + bucpslsclru + bucpslsprou	Real	0	mfcpslsu

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 5–10 MFP Cost to IP: CP Approved Data

MFP Cost Expression	Expression Type	Expression Default Value	IP Target Measure
bucpbopc	Real	0	bucpbopc
bucpbopu	Real	0	bucpbopu
bucpeopc	Real	0	bucpeopc
bucpeopu	Real	0	bucpeopu
bucpgmpv	Real	0	bucpgmpv
bucprecc	Real	0	bucprecc
bucprecu	Real	0	bucprecu
bucpslsc	Real	0	bucpslsc
bucpslsr	Real	0	bucpslsr
bucpslsu	Real	0	bucpslsu
bucpslsnetr	Real	0	bucpslsnetr

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

Table 5–11 MFP Cost to IP: OP Approved Data

MFP Cost Expression	Expression Type	Expression Default Value	IP Target Measure
buopbopc	Real	0	buopbopc
buopbopu	Real	0	buopbopu
buopeopc	Real	0	buopeopc
buopeopu	Real	0	buopeopu
buopgmpv	Real	0	buopgmpv
buoprecc	Real	0	buoprecc
buoprecu	Real	0	buoprecu
buopslsc	Real	0	buopslsc
buopslsr	Real	0	buopslsr
buopslsu	Real	0	buopslsu
buopslsnetr	Real	0	buopslsnetr

MFP Retail to IP Package

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

Scripts Used By the MFP Retail to IP Package

- mfp_to_ip_init.ksh
- mfp_to_ip_finalize.ksh

Data Mapping for MFP Retail to IP Package

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

Table 5–12 MFP Retail to IP: CP Approved Data

MFP Retail Expression	Expression Type	Expression Default Value	IP Target Measure
bucpbopc	Real	0	bucpbopc
bucpbopr	Real	0	bucpbopr
bucpbopu	Real	0	bucpbopu
bucpeopc	Real	0	bucpeopc
bucpeopr	Real	0	bucpeopr
bucpeopu	Real	0	bucpeopu
bucpigmpv	Real	0	bucpgmpv
bucpmkdclrr	Real	0	bucpmkdclrr
bucpmkdpermr	Real	0	bucpmkdpermr
bucpmkdpromor	Real	0	bucpmkdpror
bucprecc	Real	0	bucprecc
bucprecr	Real	0	bucprecr
bucprecu	Real	0	bucprecu
bucpslsclrr	Real	0	bucpslsclrr
bucpslsclru	Real	0	bucpslsclru
bucpslspror	Real	0	bucpslspror
bucpslsprou	Real	0	bucpslsprou
bucpslsregr	Real	0	bucpslsregr
bucpslsregu	Real	0	bucpslsregu
bucpslsnetr	Real	0	bucpslsnetr

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

Table 5–13 MFP Retail to IP: OP Approved Data

MFP Retail Expression	Expression Type	Expression Default Value	IP Target Measure
buopbopc	Real	0	buopbopc
buopbopr	Real	0	buopbopr
buopbopu	Real	0	buopbopu
buopeopc	Real	0	buopeopc
buocpeopr	Real	0	buocpeopr
buopeopu	Real	0	buopeopu
buopicogsc	Real	0	buopcogsc
buopigmpv	Real	0	buopgmpv
buopmkdclrr	Real	0	buopmkdclrr
buopmkdpermr	Real	0	buopmkdpermr
buopmkdpromor	Real	0	buopmkdpromor

Table 5–13 (Cont.) MFP Retail to IP: OP Approved Data

MFP Retail Expression	Expression Type	Expression Default Value	IP Target Measure
buoprecc	Real	0	buoprecc
buoprecr	Real	0	buoprecr
buoprecu	Real	0	buoprecu
buopslsclrr	Real	0	buopslsclrr
buopslsclru	Real	0	buopslsclru
buopslspror	Real	0	buopslspror
buopslsprou	Real	0	buopslsprou
buopslsregr	Real	0	buopslsregr
buopslsregu	Real	0	buopslsregu
buopslsnetr	Real	0	buopslsnetr

MFP Finalize Exports Package

There are four packages that export data from MFP Cost and MFP Retail to Assortment Planning (AP) and Item Planning (IP). These packages have two flag measures called Newly Approved. These measures indicate which data has been approved since the last export. The purpose of the MFP Finalize Exports package is to give you a way to reset these flags after MFP exports are completed.

These two flags are boolean measures at the channel/class/week level. A value of *True* indicates that the corresponding channel/class/week level data has been approved and is ready for export. One measure controls the Current Plan data and the other controls the Original Plan data.

Table 5–14 MFP Finalize Exports Measures

Measure Name	Label
mowpappcpnewb	Wp Newly Approved CP (Current Plan flag)
mowpappopnewb	Wp Newly Approved OP (Original Plan flag)

For instance, if you use MFP Retail, AP, and IP, and you approve an MFP plan for fiscal year 2011 (FY2011), this sets the Newly Approved flags to *True* in the January time frame (for the channels/classes).

If you wanted to export this plan to both AP and IP, perform the following steps:

1. Run the MFP Retail to AP Package and MFP Retail to IP Package. It does not matter which package you run first.
2. Run the MFP Finalize Exports package to reset Newly Approved flags to False.

If you do not run the MFP Finalize Exports package and later you approve an MFP plan for FY2012, the FY2011 data is still flagged as Newly Approved. During the next data export, this causes the FY2011 plan to be sent unnecessarily with the FY2012 plan. This slows the export process and may cause discrepancies since MFP exports any saved changes to the FY2011 data since the previous export.

Script Used By the MFP Finalize Exports Package

- mfp_finalize_exports.ksh

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 5–15 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 5–15 (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 5–15 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 Enabled Oracle Data Integrator 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 5–16 Loc.csv File Dimension Positions/Labels

Dimension Position/Label
STR

Table 5–16 (Cont.) Loc.csv File Dimension Positions/Labels

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

The dimensions in Table 5–16 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 Enabled Oracle Data Integrator 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.

6

Script Integration

This chapter describes the Item Planning 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 Item Planning. These scripts are located in `$IP_HOME/bin`.

Table 6–1 Integration Scripts

Application	Script Name	Arguments
AP	loadActuals.ksh	measurelist
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>

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 `$IP_HOME/bin` directory. The file name should follow the convention of AP_IP_transferList.txt.

Example
loadActuals.ksh $IP_HOME/bin/AP_IP_transferList.txt

Error Information

Table 6–3 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	Argument 'measurelist' is 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>

Table 6–4 Export to MFP Usage

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 `$IP_HOME/bin` directory. The file name should follow the convention of IP_MFP_transferList.txt.

Example
exportToMFP.ksh 1 $MFP_HOME/domain/mfpcst $IP_HOME/bin/IP_MFP_transferList.txt

Error Information

Table 6–5 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.

7

Batch Processing

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

Batch Structure Overview

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

Table 7–1 Directories Used by Batch Scripts

Directory Name	Content of the Directory
bin	Batch scripts
config	Item Planning 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 `$IP_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	informalPositionMgr.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 Item Planning. The batch environment scripts supported for Item Planning Configured for COE are described in Chapter 8, "Item Planning 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 `$IP_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=$IP_HOME/domain
- export ITEM_MASTERDOMAIN=$ITEM_DOMAINHOME/itemplan
- export ITEM_CONFIGHOME=$IP_HOME/config
- export ITEM_EXPORT=$IP_HOME/export
- export ITEM_INPUTHOME=$IP_HOME/input
- export ITEM _LOG_DIR=$IP_HOME/logs
- export ITEM _LIB=$IP_HOME/bin
- export ITEM _TEMP=$IP_HOME/temp
- export ITEM _BATCH=$IP_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:

- Calendar Hierarchy Load
- Product Hierarchy Load
- Location Hierarchy Load
- Load On Order Data
- Load Actuals Data
- Load AP Data
- Load MFP Data
- Process Actuals Data
- Run Batch Forecast
- Refresh Workbooks
- Auto Build Workbooks
- Load Mapping Measures

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. Additionally, some scripts use the Batch Script Architecture variable BSA_MAX_PROCESSES, defined in the environment.ksh script, for process parallelization.

Calendar Hierarchy Load

Script
loadhier.ksh

Usage
loadhier.ksh <hierarchy> <purgeage>

Table 7–5 Calendar Hierarchy Load Usage

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.

Example
loadhier.ksh clnd 10000

Error Information

This script utilizes standard Batch Script Architecture logging and return code processing. For more information on logging and standard return codes, see the *Oracle Retail Batch Script Architecture Implementation Guide*.

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>

Table 7–6 Product Hierarchy Load Usage

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.

Example
loadhier.ksh prod 10000

Error Information

This script utilizes standard Batch Script Architecture logging and return code processing. For more information on logging and standard return codes, see the *Oracle Retail Batch Script Architecture Implementation Guide*.

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>

Table 7–7 Location Hierarchy Load Usage

Argument	Description	Notes
hierarchy	Sets the RPAS name of the hierarchy to be loaded.	The default for the product hierarchy is loc.

Table 7–7 (Cont.) Location Hierarchy Load Usage

Argument	Description	Notes
purgeage	Sets the number of days after updating a hierarchy position before purging the position.	The default is 10000.

Example
loadhier.ksh loc 10000

Error Information

This script utilizes standard Batch Script Architecture logging and return code processing. For more information on logging and standard return codes, see the *Oracle Retail Batch Script Architecture Implementation Guide*.

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>

Table 7–8 Load On Order 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 `$IP_HOME/bin` directory. The file name should follow this convention: `AP_IP_transferList.txt`.

Control File
LoadOnOrderList.txt: Contains these measures that can be loaded as part of this script:

- ipwponordc
- ipwponordr
- ipwponordu

Example
loadActuals.ksh LoadOnOrderList.txt

Error Information

Table 7–9 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	Argument 'measurelist' is 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>

Table 7–10 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 `$IP_HOME/bin` directory. The file name should follow this convention: `AP_IP_transferList.txt`.

Control File
LoadActualsList.txt: Contains these measures that can be loaded as part of this script:

drtyeopclrc	drtyrecu	drtyslsregr
drtyeopclrr	drtyslsclrc	drtyslsregu
drtyeopclru	drtyslsclrr	drwpdefslscrvp
drtyeopregc	drtyslsclru	drwpdefslscrvtx
drtyeopregr	drtyslsproc	ipwpcutkpsswv

drtyeopregu	drtyslspror	ipwportkpssv
drtymkdar	drtyslsprou	ipwpprcopsswv
drtyrecc	drtyslsregc	ipwpedwkdv
drtyrecr		

Example
loadActuals.ksh LoadActualsList.txt

Error Information

Table 7–11 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	Argument 'measurelist' is 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>

Table 7–12 Load AP 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 `$IP_HOME/bin` directory. The file name should follow this convention: `AP_IP_transferList.txt`.

Control File
AP_IP_transferList.txt: Contains these measures that can be loaded as part of this script:

APCpAsgn2StrB	APCpMkdPermR	APCpSizGrpTx
APCpBOPC	APCpMkdProR	APCpSizRangeTx
APCpBOPR	APCpPckSzU	APCpSlsClrC
APCpBOPU	APCpPerfGrpTx	APCpSlsClrR
APCpCOGSC	APCpPrcClrC	APCpSlsClrU
APCpDelFreqU	APCpPrcClrR	APCpSlsRegC
APCpEOPC	APCpPrcIniR	APCpSlsRegR
APCpEOPR	APCpPrcProR	APCpSlsRegU
APCpEOPU	APCpPrcR	APCpSlsStartD
APCpExitD	APCpPresMinU	APCpStrU
APCpFabricTx	APCpRecC	APCpVendorTx
APCpItmAt1MapTx	APCpRecR	APCpWeeksClrU
APCpItmAt2MapTx	APCpRecU	APCpWeeksRegU
APCpMkdClrR	APCpSftyStckU	

Example
loadActuals.ksh AP_IP_transferList.txt

Error Information

Table 7–13 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	Argument 'measurelist' is 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 MFP Data

Script
loadActuals.ksh

Usage
loadActuals.ksh <measurelist>

Table 7–14 Load AP 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 `$IP_HOME/bin` directory. The file name should follow this convention: `MFP_IP_transferList.txt`.

Control File
MFP_IP_transferList.txt: Contains these measures that can be loaded as part of this script.

> **Note:** The control file contains measures for both MFP Retail and MFP Cost. Each variation of MFP contains measures that the other variation does not contain. As a result, you may want to use only the measure set (MFP Retail or MFP Cost) that is applicable to your installation. For the individual measure lists, see MFP Retail Measure List and MFP Cost Measure List.

bucpbopc	bucpslsnetr	buopmkdpermr
bucpbopr	bucpslsproc	buopmkdpror
bucpbopu	bucpslspror	buoprecc
bucpcogsc	bucpslsprou	buoprecr
bucpeopc	bucpslsr	buoprecu
bucpeopr	bucpslsregc	buopslsc
bucpeopu	bucpslsregr	buopslsclrc
bucpgmpv	bucpslsregu	buopslsclrr
bucpmkdclrr	bucpslsu	buopslsclru
bucpmkdpermr	buopbopc	buopslsnetr
bucpmkdpror	buopbopr	buopslsproc
bucprecc	buopbopu	buopslspror
bucprecr	buopcogsc	buopslsprou
bucprecu	buopeopc	buopslsr
bucpslsc	buopeopr	buopslsregc
bucpslsclrc	buopeopu	buopslsregr
bucpslsclrr	buopgmpv	buopslsregu
bucpslsclru	buopmkdclrr	buopslsu

MFP Retail Measure List

bucpbopc	bucpslsproc	buopmkdpror
bucpbopr	bucpslspror	buoprecc
bucpeopc	bucpslsprou	buoprecr
bucpeopr	bucpslsregc	buopslsc
bucpgmpv	bucpslsregr	buopslsclrc
bucpmkdclrr	bucpslsregu	buopslsclrr
bucpmkdpermr	buopbopc	buopslsclru
bucpmkdpror	buopbopr	buopslsproc
bucprecc	buopeopc	buopslspror
bucprecr	buopeopr	buopslsprou
bucpslsclrc	buopgmpv	buopslsregc
bucpslsclrr	buopmkdclrr	buopslsregr
bucpslsclru	buopmkdpermr	buopslsregu

MFP Cost Measure List

bucpbopc	bucpslsnetr	buoprecc
bucpbopu	bucpslsu	buoprecu
bucpeopc	buopbopc	buopslsc
bucpeopu	buopbopu	buopslsnetr
bucpgmpv	buopcogsc	buopslsproc
bucprecc	buopeopc	buopslsprou
bucprecu	buopeopu	buopslsr
bucpslsc	buopgmpv	buopslsu

Example
loadActuals.ksh MFP_IP_transferList.txt

Error Information

Table 7–15 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	Argument 'measurelist' is 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

This script utilizes standard Batch Script Architecture logging and return code processing. For more information on logging and standard return codes, see the *Oracle Retail Batch Script Architecture Implementation Guide*.

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

This script utilizes standard Batch Script Architecture logging and return code processing. For more information on logging and standard return codes, see the *Oracle Retail Batch Script Architecture Implementation Guide*.

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

Error Information

This script utilizes standard Batch Script Architecture logging and return code processing. For more information on logging and standard return codes, see the *Oracle Retail Batch Script Architecture Implementation Guide*.

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 Item Planning 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

Error Information

This script utilizes standard Batch Script Architecture logging and return code processing. For more information on logging and standard return codes, see the *Oracle Retail Batch Script Architecture Implementation Guide*.

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 with the domain before running this script. See the *Oracle Retail Predictive Application Server User Guide for the Classic Client* 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>

Table 7–16 Load Mapping Measures 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 `$IP_HOME/bin` directory. The file name should follow this convention: `AP_IP_transferList.txt`.

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

Example
loadActuals.ksh LoadMapList.txt

Error Information

Table 7–17 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	Argument 'measurelist' is 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.

8

Item Planning Configured for COE

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

For information on installing COE, see the Oracle Retail Item Planning Configured for COE (IP COE) documentation.

Integration Methods and Communication Flow

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

Data Exported from COE to Item Planning Configured for COE

Optimized forecasts are extracted from COE into pipe-delimited flat files (with the exception of Pricing Group Mapping, which is tab-delimited). 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 COE.

- Markdown Recommendations
- Forecast Summaries
- Forecast Activities
- Pricing Group Mapping

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 8–1 describes the layout of this file.

Table 8–1 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

Table 8–1 (Cont.) Layout of the Markdown Recommendations File

Field Name	Data Type	Maximum Length	Nullable?	Field Description
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 Item Planning, the day level information id aggregated during the load process. Table 8–2 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 8–2 Markdown Recommendations File After Transformation

Measure Name	Measure Label	File Name	Notes
IpWpCoeRCMdPDV	COE Recommended Item Clearance Markdown Retail Price	coeitmmkdprc.csv.rpl	Aggregation Type of average_pop
IpWpCoeRCMdPDU	COE Recommended Pricing Group Clearance Markdown Retail Price	coepgmkdprc	

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

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

Forecast Summaries

The forecast summaries file, mdo-rpas-forecast-summ.txt, contains forecasted opportunity costs for the individual style-colors or price groups. Table 8–3 describes the layout of this file.

Table 8–3 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
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 Item Planning 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 8–4 provides details on the file.

Table 8–4 Forecast Summaries File After Transformation

Measure Name	Measure Label	File Name
IPWPCORCOPCOV (COE Recommended Opportunity Cost (measure 94 - Measure Analysis))	COE Forecasted Item Opportunity Cost	coeitmoppcost.csv.rpl
IPWPCORCOPOU (COE Recommended Pricing Group Opportunity Cost)	COE Forecasted Pricing Group Opportunity Cost	coepgoppcost.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 8–5 describes the layout of this file.

Table 8–5 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
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 8–6 provides details on the files.

Table 8–6 Forecast Activities File After Transformation

Measure Name	Measure Label	File Name
IpWpCoRcFcSULV	COE Recommended Forecasted Item Sales Unit	coeitmslsu.csv.rpl
IpWpCoPrSlsArV	COE Forecasted Item Sales Price	coeitmslsprc.csv.rpl
IPWPCOPEOPIUV	COE Forecasted Item Inventory Units	coeitminvu.csv.rpl
IPWPCORCMDPPV	COE Forecasted Clearance Markdown Retail Price Plan	coeitmtkdprc.csv.rpl
IpWpCoRcFcSULU	COE Recommended Forecasted Pricing Group Sales Units Loaded	oepgslsu.csv.rpl
IpWpCoPrSlsArU	COE Forecasted Pricing Group Sales Price	coepgslsprc.csv.rpl
IPWPCOPEOPIUU	COE Forecasted Pricing Group EOP Inventory Units	coepginvu.csv.rpl
IPWPCORCMDPPU	COE Forecasted Pricing Group Clearance Markdown Retail Price Plan	coepgtkdprc.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

coepgslsu.csv.rpl
20041009,10100004769639001,U#C1,149.95
20041016,10100004769639001,U#21,149.95

coepgslsprc.csv.rpl
20041009,10100004769639001,U#C1,149.95
20041016,10100004769639001,U#21,149.95

coepginvu.csv.rpl
20041009,10100004769639001,U#C1,149.95
20041016,10100004769639001,U#21,149.95

coepgtkdprc.csv.rpl
20041009,10100004769639001,U#C1,149.95
20041016,10100004769639001,U#21,149.95

Pricing Group Mapping

The pricing group mapping file, pg.txt, contains the product/location to pricing group mapping information.

Note: This file is tab-delimited.

Table 8–7 describes the layout of this file.

Table 8–7 Layout of the Pricing Group Mapping 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
Pricing Group ID	String	80	No	Client pricing group unique identifier.
Pricing Group Collection	String	80	No	Client pricing group description.
Pricing Group First Effective Date	Date	9	No	Starting date at which pricing group is effective.
Pricing Group Last Effective Date	Date	9	Yes	Ending data until which the pricing group is valid. Null means no last date defined.

The following is an example of the content of the Pricing Group Mapping file:

01001049885337U#U301001049885 FU_1_26 UACME TAHOE DOWN VEST FU_1_2618-FEB-10

01001049885492U#U301001049885 FU_1_26 UACME TAHOE DOWN VEST FU_1_2618-FEB-10

01001049885212U#U301001049885 FU_1_19 UACME TAHOE DOWN VEST FU_1_1918-FEB-10

To load this file into RPAS, the file needs to be transformed into a CSV file. The order of the hierarchy positions is product followed by location. The effective date of the mapping is ignored at this time.

The following measure is loaded from the resulting CSV file.

Table 8–8 Pricing Group Mapping File After Transformation

Measure Name	Measure Label	File Name	Notes
IpWppginfotx	Pricing Group	ipwppginfotx.csv.rpl	Two-dimensional string measure at skup/pzon

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

ipwppginfotx.csv.rpl
01001049885337,U#U3,ACME TAHOE DOWN VEST/FU_1_26
01001049885492,U#U3,ACME TAHOE DOWN VEST/FU_1_26

Data Exported from Item Planning on RPAS to COE

Business rules are exported from Item Planning on RPAS to COE.

Business Rules

COE supports the setting of business rules that drive the optimization process. Within the Item Planning 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 is an abbreviation for 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 Item Planning 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, [clnd].[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 8–9 describes the specification for the ash_brm_instance_tbl.dat COE file.

Table 8–9 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 Item Planning, 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:

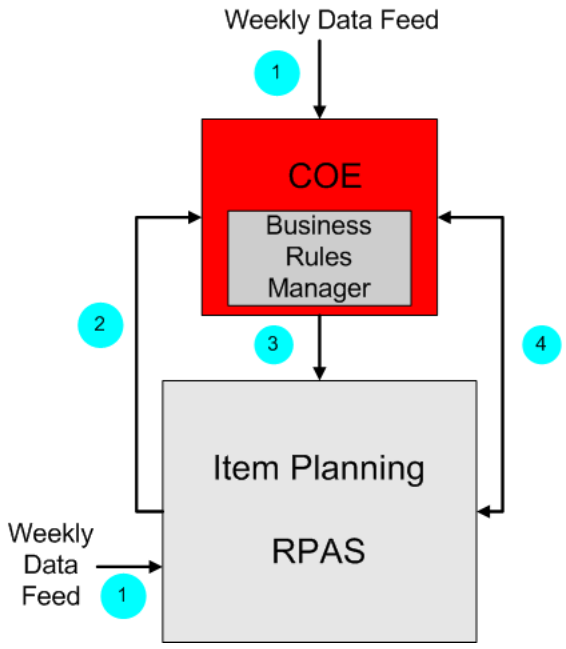
- The file must become pipe (|) delimited.

- The first field is equal to the style-color ID that is exported.
- The second field is equal to style-color.
- The third field is equal to the price zone ID that is exported.
- The fourth field is equal to price zone.
- The fifth field is equal to the corresponding Rule_Name.
- The sixth field is the measure value that is exported.
- The seventh and eighth fields is null at this point.
- The ninth field is set to zero.

Data Flow Between COE and IP COE

Figure 8–1 illustrates the flow of data between COE and IP COE. 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 COE



Data Flow Description

The following scripts are provided:

- `rpas2coe.sh`: RPAS provides this script for exporting and transforming data sent to COE.
- `coe2rpas_initial.sh`: RPAS provides this script to transform and load initial business rules that are present in the COE extracts to IP COE. Retailers are responsible for ensuring that the files are placed in the `$IP_HOME/bin/fromCOE` folder at domain build time.

  This utility is typically run at build time; therefore it is part of the build script. After the initial business rules are loaded, they are typically modified in IP COE and sent to COE using `rpas2coe.sh` script.

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 COE. The weekly feeds into COE and IP COE 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 COE on a nightly or weekly basis. Transformation of the files is done.
3. IP COE 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 COE 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 COE, a user can enter in What-If information. This information is sent to COE through the COE special expression for optimization. COE returns revised optimization information like forecasted sales, inventory, and markdowns. This is all real-time interaction.

RPAS Special Expression for COE

The communication between IP COE 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 Item Planning 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 special expression supports the Oracle WebLogic Server (OWL), version 10.3.2.

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.

An auto login wallet file is created using Credential Store Manager (CSM) framework. Secret store entries are added to this wallet file for keystore, private key(s) and COE user.

- **csmwallet_mapname Measure**

This measure stores the auto login wallet map name. Run the following mace command to set the measure value to auto login wallet map name:

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

In the above expression, XXX is the map name of the auto login wallet. All other arguments are values defined within the COE application.

- **csmwallet_file Measure**

  This measure stores the auto login wallet file name. Run the following mace command to set the measure value to auto login wallet file name:

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

  In the above expression, XXX is the file name of the auto login wallet file. All other arguments are values defined within the COE application.

- **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 save_credential.sh script to create a wallet entry keyadm:

     save_credential.sh -a *keyadm* -u *keyadm* -p *csmwallet_mapname* -l *csmwallet_file password*

     Replace the credentials in italics with actual csmwallet_mapname and csmwallet_file, keystore admin name 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.

Authentification An IP COE system user account will be created on both the 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 IP COE 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 save_credential.sh script to create a wallet entry for system user:

   save_credential.sh -a *coe_system_user* -u *coe_system_user* -p *csmwallet_mapname* -l *csmwallet_file password*

   Replace the credentials in italics with actual csmwallet_mapname, csmwallet_file, system user name 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 Item Planning. 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 save_credential.sh script to create a wallet entry for keyalias_enc :

   save_credential.sh -a *keyalias_enc* -u *keyalias_enc* -p *csmwallet_mapname* -l *csmwallet_file password*

   Replace the credentials in italics with actual csmwallet_mapname, csmwallet_file, keyalias_enc user name and password. The password for the alias should be identical in keystore and in wallet.

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 Item Planning. This measure stores the keystore alias that will be used for providing certificates for client side signing. The alias must be registered in the kesytore specified by the keystore_file measure.

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

1. Use save_credential.sh script to create a wallet entry for keyalias_sig :

   save_credential.sh -a *keyalias_sig* -u *keyalias_sig* -p *csmwallet_mapname* -l *csmwallet_file password*

   Replace the credentials in italics with actual csmwallet_mapname, csmwallet_file, keyalias_sig user name and password. The password should be identical in keystore and in wallet.

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.

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 OWL require the following jar files in the given order:

1. $RPAS_HOME/lib/oracleRpasUtils.jar
2. $IP_HOME/java/lib/coeexpression_owl.jar
3. $IP_HOME/java/lib/ant.jar
4. $IP_HOME/retailpublicsecurityapi/lib/retail-public-security-api.jar
5. com.bea.core.descriptor.wl.binding_1.3.0.0.jar
6. com.bea.core.descriptor.wl_1.3.0.0.jar
7. com.bea.core.xml.beaxmlbeans_1.1.0.0_2-4-1.jar
8. com.bea.core.xml.staxb.buildtime_1.4.1.0.jar
9. com.bea.core.xml.xmlbeans_1.1.0.0_2-4-1.jar
10. glassfish.jaxws.rt_1.1.0.0_2-1-4.jar
11. javax.xml.bind_2.1.1.jar
12. javax.xml.ws_2.1.1.jar
13. weblogic.jar
14. webservices.jar
15. wlfullclient.jar
16. xqrl.jar
17. com.bea.core.xml.weblogic.xpath_1.4.0.0.jar

Java libraries 4-16 should be obtained from Weblogic server.

Libraries 5,6,7,8,9,10,11,12 and17 can be obtained when you do unzip the $WEBLOGIC_HOME/server/lib/wseeclient.zip file.

Libraries 13, 14 and 16 can be copied from $WEBLOGIC_HOME/server/lib/ directory.

Libraries wlfullclient.jar should be built from $WEBLOGIC_HOME/server/lib directory using "java -jar wljarbuilder.jar" command.

The Weblogic Java libraries (5~17) would typically be copied into $IP_HOME/javatools/lib directory.

This can be set on Unix as follows:

```
export RPAS_JAVA_CLASSPATH="$RPAS_HOME/lib/oracleRpasUtils.jar;$IP_
HOME/java/lib/coeexpression_owl.jar;$IP_
HOME/javatools/lib/com.bea.core.descriptor.wl.binding_1.4.0.0.jar;$IP_
HOME/javatools/lib/com.bea.core.descriptor.wl_1.4.0.0.jar;$IP_
HOME/javatools/lib/com.bea.core.xml.beaxmlbeans_2.5.0.0_2-5-1.jar;$IP_
HOME/javatools/lib/com.bea.core.xml.staxb.buildtime_1.6.0.0.jar;$IP_
HOME/javatools/lib/com.bea.core.xml.xmlbeans_2.2.0.0_2-5-1.jar;$IP_
HOME/javatools/lib/glassfish.jaxws.rt_1.3.0.0_2-1-5.jar;$IP_
HOME/javatools/lib/javax.xml.bind_2.1.1.jar;$IP_HOME/javatools/lib/javax.xml.ws_
2.1.1.jar;$JAVA_HOME/lib/tools.jar;$IP_HOME/javatools/lib/weblogic.jar;$IP_
HOME/javatools/lib/webservices.jar;$IP_HOME/javatools/lib/wlfullclient.jar;$IP_
HOME/javatools/lib/xqrl.jar;$IP_
HOME/javatools/lib/com.bea.core.xml.weblogic.xpath_1.5.0.0.jar;$IP_
HOME/retailpublicsecurityapi/lib/retail-public-security-api.jar;$IP_
```

```
HOME/java/lib/ant.jar;$RPAS_JAVA_CLASSPATH"
```

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 Item Planning. 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 8–10 and Table 8–11. 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 8–10 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 8–10 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.

Table 8–10 (Cont.) COE Special Expression Input Arguments

Name	Type	BaseIntx	COE Semantics	Description
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 8–11 describes the output arguments for the COE special expression.

Table 8–11 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 clearance markdown retail price plan.
recperoff	Double	Base Intx	forecastResponse.forecasts.mkdnRec.recPerOff	Revised recommended price percent off original price.

Table 8–11 (Cont.) COE Special Expression Output Arguments

Name	Type	BaseIntx	COE Semantics	Description
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 `$IP_HOME` directory. The extracted files are pipe-delimited flat files.

Table 8–12 Directories Used by Batch Scripts

Directory Name	Content of the Directory
bin	Batch scripts
config	Item Planning 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 Item Planning. Both scripts reside in the `$IP_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 `$IP_HOME/bin/fromCOE` directory are transformed and put into the input directory of the IP COE domain.
- `rpas2coe.sh` is used for exporting and transforming IP COE data into COE compatible data. The IP COE files are exported into the `$IP_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 `$IP_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 8–13 lists information on the daily batch scripts.

Table 8–13 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 8–14 lists information on the weekly batch scripts.

Table 8–14 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	informalPositionMgr.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

Table 8–14 (Cont.) Weekly Batch Scripts

Description	Script Name	Batch Directory	Dependency
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
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 8–15 lists information on the unscheduled administration script. This script is located in the `$IP_HOME/bin` directory.

Table 8–15 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 Item Planning 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 `$IP_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=$IP_HOME/domain
- export ITEM_MASTERDOMAIN=$ITEM_DOMAINHOME/itemplan
- export ITEM_CONFIGHOME=$IP_HOME/config
- export ITEM_EXPORT=$IP_HOME/export
- export ITEM_INPUTHOME=$IP_HOME/input
- export ITEM _LOG_DIR=$IP_HOME/logs
- export ITEM _LIB=$IP_HOME/bin
- export ITEM _TEMP=$IP_HOME/temp
- export ITEM _BATCH=$IP_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>

Table 8–16 Load COE Batch 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 `$IP_HOME/bin` directory.

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

Error Information

Table 8–17 Load Actuals Data Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	Argument 'measurelist' is 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.