Oracle® Retail Merchandise Financial Planning

Operations Guide

Release 14.1.3

E70284-01

July 2016

ORACLE®

Oracle Retail Merchandise Financial Planning Operations Guide, Release 14.1.3

E70284-01

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Primary Author: Bernadette Goodman

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 Fashion Planning Bundle Integration

5   Script Integration

6   RMS and Retail Analytics Integration

7   Batch Processing

List of Tables

List of Figures

Send Us Your Comments

Oracle Retail Merchandise Financial Planning Operations Guide, Release 14.1.3

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

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

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

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

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

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

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

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

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

Preface

This Operations Guide provides critical information about the processing and operating details of Oracle Retail Merchandise Financial 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 Merchandise Financial 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 Merchandise Financial Planning Release 14.1.3 documentation set:

- *Oracle Retail Merchandise Financial Planning Cost Installation Guide*
- *Oracle Retail Merchandise Financial Planning Release Notes*
- *Oracle Retail Merchandise Financial Planning Retail Installation Guide*

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

- Oracle Retail Item Planning documentation
- Oracle Retail Item Planning Configured for COE documentation
- Oracle Retail Assortment Planning documentation
- Oracle Retail Size Profile Optimization 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, 14.1) or a later patch release (for example, 14.1.3). 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

Oracle Retail product documentation is 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. You can obtain them through My Oracle Support.)

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 Merchandise Financial Planning (MFP) provides strategic and financial product planning functions. These functions support industry planning standards for pre-season and in-season processes. For a more detailed overview of the functionality within MFP, see the *Oracle Retail Merchandise Financial Planning Cost User Guide* and *Oracle Retail Merchandise Financial Planning Retail User Guide*.

Contents of this Guide

This implementation guide addresses the following topics:

- Chapter 1, "Introduction". Overview of the MFP 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 MFP RPAS domain.

- Chapter 4, "Fashion Planning Bundle Integration". Overview of the Fashion Planning Bundle data flow and integration.

- Chapter 5, "Script Integration". Overview of the script integration used for the Fashion Planning Bundle.

- Chapter 6, "RMS and Retail Analytics Integration". Overview of RMS integration as well as integration with data warehouses and markdown optimization applications.

- Chapter 7, "Batch Processing". Explanation of batch scheduling and batch designs.

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 4, "Fashion Planning Bundle Integration" and Chapter 5, "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
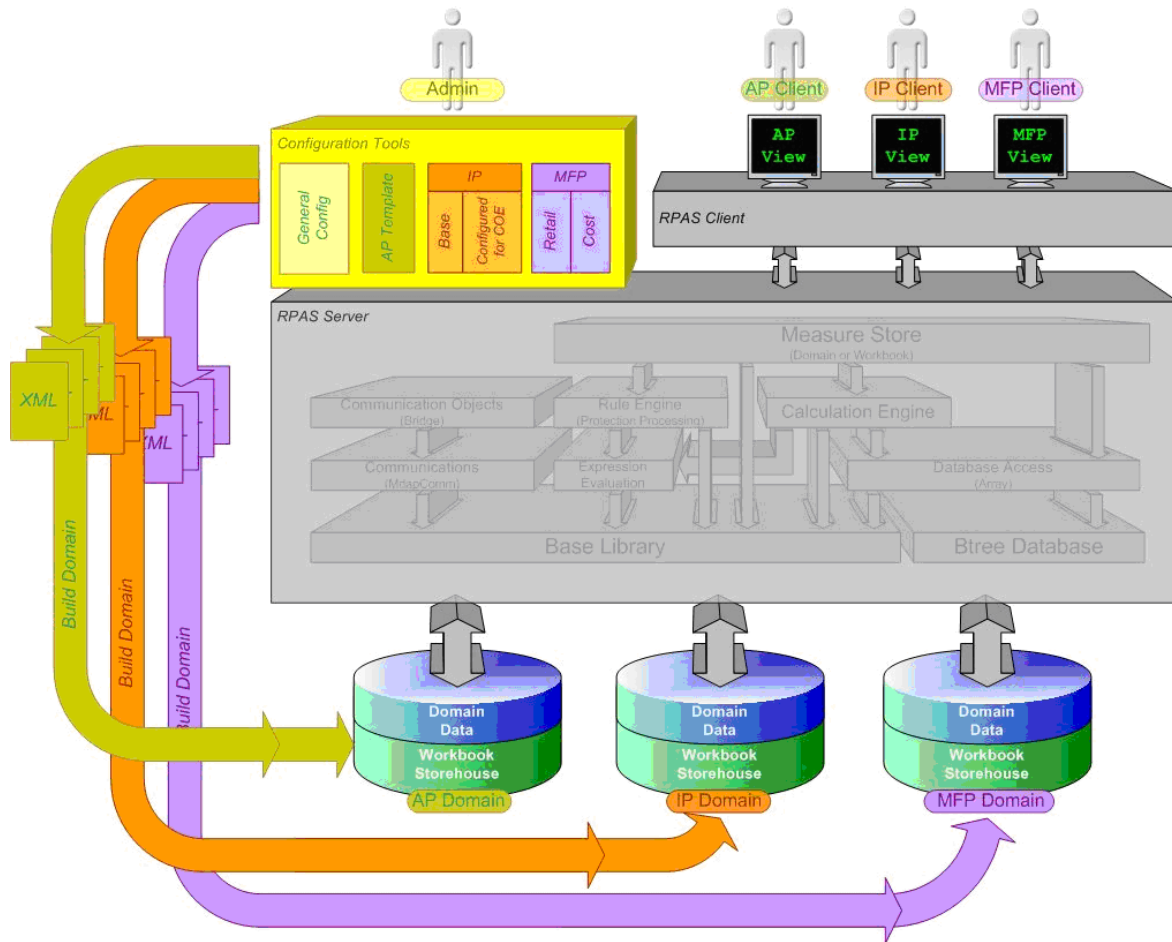
This diagram describes the RPAS template applications. In the actually scenario, these templates are not applications in the same way that the RPAS client is an application since end-users are not presented a user interface specific to the template. The templates are pre-defined means by which to view 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 through 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 MFP.

Figure 1–2   Business Process Workflow
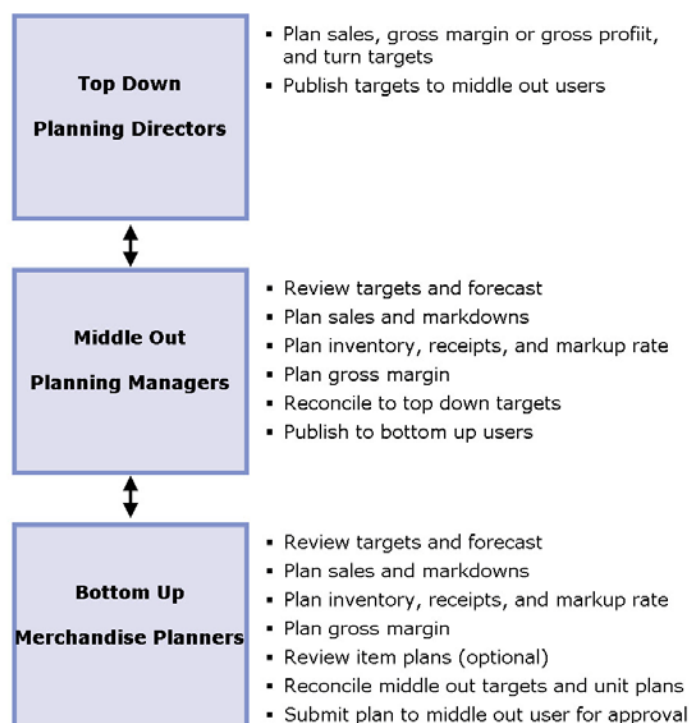
Key Features of MFP

MFP provides the following features:

- Setting and passing targets
- Creating a financial plan in a structured method
- Plan reconciliation
- Plan approval
- Plan maintenance
- Replanning
- Plan monitoring

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 Merchandising System (RMS)
- Oracle Retail Markdown Optimization (MDO)

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 MFP
- Retailer's hierarchical (SKU/store/day) data
- MFP 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 MFP:

- 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 MFP targets. Less data can be used, but the more data that is available, the more statistical significance can be given to the MFP targets.

It is also important to have two years of history to produce an optimal forecast. Both MFP Cost and MFP Retail use the embedded forecast.

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. Although the lowest level of planning within MFP is SubClass, data is loaded at the style-color level.

- Store—number of physical, web, and other distinct retail outlets. Although store is not the lowest level of planning, data may be loaded at that 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 causes contention.

How data is partitioned has an impact on the business process. The MFP 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, separate domain could be provided for Merchandise of Men, Women, and Children. So that when a user commits a data in the men's merchandise domain, does not affect the users in the women's or children's domains because of the use of partitioning of the domains.

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 MFP 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 the *Oracle Retail Predictive Application Server User Guide for the Fusion Client*.

Plug-ins

> **Note:** There is no Configuration Tools plug-in for MFP.

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 MFP RPAS domain:

- Changes to the code in the RPAS libraries

  The configuration is not affected by this type of patch. For these types of changes, applying the patch is a straightforward process.

- Changes to the configuration

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

  > **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, see 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 targets. 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 re-create targets needs to be determined.

- You have to consider at what interval to load the latest sales and inventory data. A weekly load of transactional type data is supported since the base intersection is at week. It is recommended that the information coming from other Fashion Bundle application be loaded daily.

- Product availability and seasonal changes can be reasons for recalculating the targets. It can also be triggered by the addition of new products and availability of substantial new sales and inventory history.

Security

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

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

- World—Allow any user to open and edit the workbook.

- Group—Allow only those users in their same group to open and edit the workbooks.

- User—Allow no other users to open and edit the workbook.

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

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

Installation Dependencies

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

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

- For information on installing MFP, see the *Oracle Retail Merchandise Financial Planning Retail Installation Guide* and the *Oracle Retail Merchandise Financial Planning Cost 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 MFP_COST_HOME for MFP Cost and MFP_RETAIL_HOME for MFP Retail. The UNIX user performing the installation needs to set up an environmental variable called MFP_COST_HOME or MFP_RETAIL_HOME in the user's profile:

export MFP_COST_HOME=<*full path name to MFP Cost home*>

or

export MFP_RETAIL_HOME=<*full path name to MFP Retail home*>

The MFP_Home is now referred either as $MFP_COST_HOME or $MFP_RETAIL_ HOME depending on the environment.

> **Note:** In this document, to simply MFP Cost or MFP Retail is referred as $MFP_HOME depending on the installation. All reference to $MFP_HOME is interpreted either as $MFP_COST_HOME or $MFP_RETAIL_HOME appropriately.
>
> The bath scripts for MFP Cost and MFP Retail do not use the variable $MFP_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, see the *Oracle Retail Predictive Application Server Installation Guide*.

MFP Installation

The MFP installer performs the following functions:

- Downloads the configuration and batch scripts into the `$MFP_HOME/config` and `$MFP_HOME/bin` directories.

- Downloads a set of sample hierarchy and data files into the `$MFP_HOME/input` directory.

- Builds a sample domain at:

  - For MFP Cost, `$MFP_HOME/domain/mfpcst`

  - For MFP Retail, `$MFP_HOME/domain/mfprtl`

Custom Domain Build

To do a custom build of a domain:

1. Change to the configuration directory: `cd $MFP_HOME/config`.

2. Update the globaldomainconfig.xml file with the correct domain paths. In addition, the groups that were part of the sample installation are replaced with the position information for the groups that are part of the retailer's hierarchy.

3. Change to the bin directory: `cd $MFP_HOME/bin`.

4. If needed, update the default environment variables in environment.ksh.

5. Execute the build_mfp.ksh script: `./build_mfp.ksh`.

MFP Taskflow for the RPAS Fusion Client

The MFP 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 Fusion Client*.

Handling Common Hierarchy Files in the Fashion Planning Bundle Applications

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 AP, IP, MFP, and SPO.

> **Note:** The common hierarchy is not available for IP Configured for COE.

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

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

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

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

For more information on RPAS utilities, 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*.

Building the MFP Domain

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

Table 3–1 lists the build scripts. These scripts are located in the $MFP_HOME/bin directory.

Table 3–1   Build Scripts

Description	Script Name	Dependency
Build a domain.	build_mfp.ksh	None
Patch a domain when the configuration changes do not affect formatting.	patch_mfp_keepformats.ksh	None
Patch a domain when the configuration changes require updates to formatting.	patch_mfp_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_mfp.ksh

Usage
build_mfp.ksh

Error Information

Table 3–2   Building a Domain Error Information

Task Name	Error Code	Abort Required?	Description of Error
build_mfp	1	yes	Error during domain build.
build_mfp	3	yes	Error when loading measures.
build_mfp	4	yes	Error in the master domain calculations.
build_mfp	6	yes	Error in local domain calculations.
build_mfp	7	yes	Error when disabling commit later and insert measure functionality.
build_mfp	8	yes	Error when running copyDomain to set 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 Classic Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Fusion 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 Classic Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Fusion Client* for details on these utilities.

- All hierarchy and measure files need to be placed in $MFP_HOME/input before the execution of this script.

- The commit later and insert measure functionality is disabled.

Patching a Domain

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

Script
patch_mfp_keepformats.ksh
or
patch_mfp_deleteformats.ksh

Usage
patch_mfp_keepformats.ksh
patch_mfp_deleteformats.ksh

Error Information

Table 3–3   Patching a Domain Error Information

Task Name	Error Code	Abort Required?	Description of Error
patch_mfp_deleteformats	10	yes	Error during domain patch.
patch_mfp_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 utilize the Configuration Tools rpasInstall utility to build a domain. 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* 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 Classic Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Fusion Client*.

Loading and Extracting Data

Data is loaded into MFP using the standard RPAS approach. 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* for details on formatting the load data files and on the utilities that enable administrators to load data into RPAS. For more information on data integration among the Fashion Planning Bundle applications, see Chapter 4, "Fashion Planning Bundle Integration". If you are using script integration for the bundle, see Chapter 5, "Script Integration". For information about RMS integration, see Chapter 6, "RMS and Retail Analytics Integration". For information on batch scripts, see Chapter 7, "Batch Processing".

4

Fashion Planning Bundle Integration

This chapter describes the overall flow of data and integration among the Fashion Planning Bundle applications.

For information about script integration for the Fashion Planning Bundle, see Chapter 5, "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: Item Planning (IP), Clearance Optimization Engine (COE), Assortment Planning (AP), Merchandise Financial Planning (MFP), and Size Profile Optimization (SPO).

Figure 4–1 shows the conceptual overview of the integration of these products.

Figure 4–1   Conceptual Overview
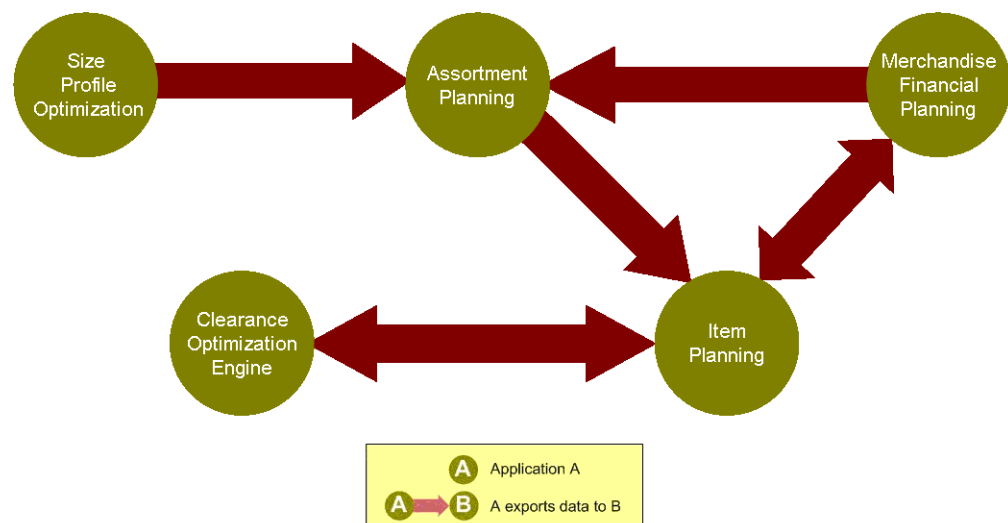
This solution supports data sharing among these applications. Note that the data sharing functionality is not dependent on the presence of all these applications. The defined data sharing between any of the applications works for the entire suite as well as for a subset of the applications.

Integration Interface Data Flow Description

These descriptions explain each of the data flows in Figure 4–1.

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. 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 4–2 shows the applications and data flow that are part of the Fashion Planning Bundle.

Figure 4–2   Overview of Fashion Planning Bundle Integration
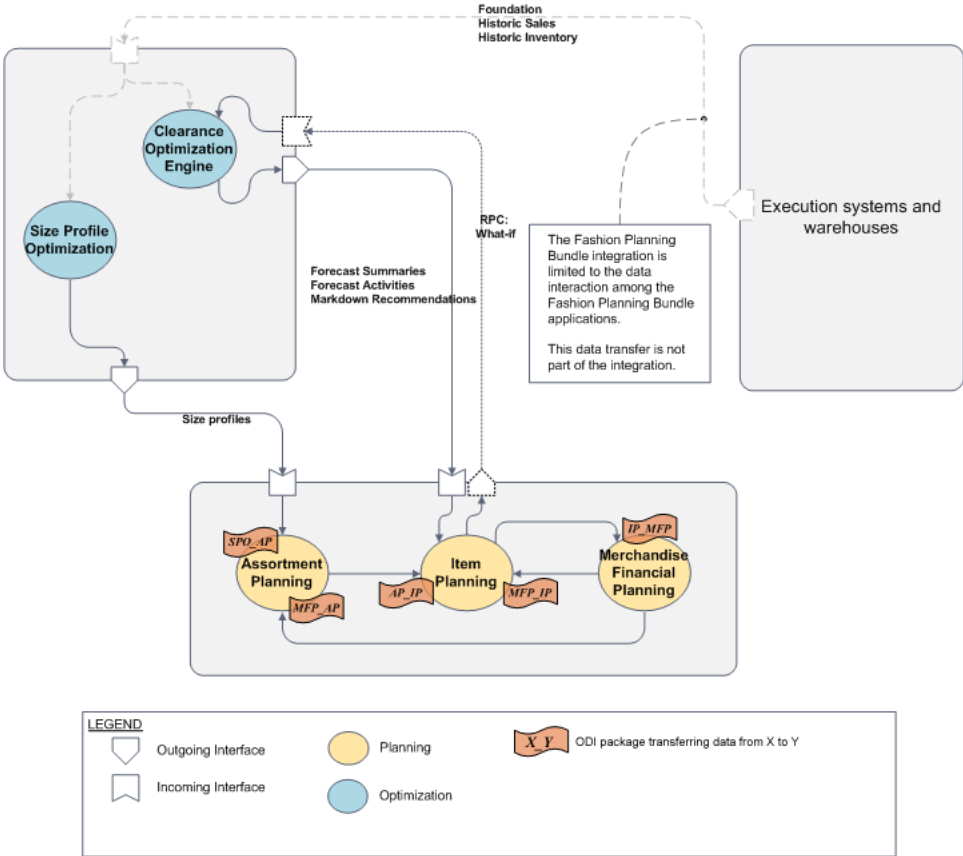
The applications shown in the diagram are in the following categories:

- 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 customer in-house 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.

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

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 4–1   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 4–2   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
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 4–3   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 4–4   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
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 4–5   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 4–6   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 4–7   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
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

Table 4–7 (Cont.) AP to IP Data

AP Expression	Expression Type	Expression Default Value	IP Target Measure
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 4–8 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 4–9 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

MFP Retail Expression	Expression Type	Expression Default Value	AP Target Measure
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 4–10   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 4–11   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

Table 4–11 (Cont.) MFP Cost to IP: OP Approved Data

MFP Cost Expression	Expression Type	Expression Default Value	IP Target Measure
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 4–12 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

Table 4–12   (Cont.) MFP Retail to IP: CP Approved Data

MFP Retail Expression	Expression Type	Expression Default Value	IP Target Measure
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 4–13   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
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 4–14   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

5

Script Integration

This chapter describes the script integration of the Fashion Planning Bundle applications.

For information about the integration for the Fashion Planning Bundle, see Chapter 4, "Fashion Planning Bundle Integration". For information about RMS integration, see Chapter 6, "RMS and Retail Analytics Integration".

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 a 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, MFP_AP_transferList.txt includes the measure names exported from MFP to AP.

  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 5–1 lists the integration scripts for MFP. These scripts are located in `$MFP_HOME/bin`.

Table 5–1   Integration Script

Application	Script Name	Arguments
MFP	loadactuals.ksh	measurelist, maxprocesses
IP	exportToIP.ksh	maxprocesses, destination, measurelist
AP	exportToAP.ksh	maxprocesses, destination, measurelist
IP and AP	resetExportFlags.ksh	none

Batch Designs

This section contains detailed information on the following integration scripts:

- Load Actuals
- Export to IP
- Export to AP
- Reset Export Flags

Load Actuals

Script
loadactuals.ksh

Usage
loadactuals.ksh <measurelist> <maxprocesses>

Table 5–2   Load Actuals 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 `$MFP_HOME/bin` directory. The file name should follow the convention of IP_MFP_transferList.txt.
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.

Control File
IP_MFP_transfer_list.txt contains the following list of measures that can be loaded as part of this script:

MFP Cost

List of Measures		
ipcpbopc	ipcpslsc	ipopgmpv
ipcpbopu	ipcpslsr	ipoprecc
ipcpeopc	ipcpslsu	ipoprecu
ipcpeopu	ipopbopc	ipopslsc
ipcpgmpv	ipopbopu	ipopslsr

List of Measures		
ipcprecc	ipopeopc	ipopslsu
ipcprecu	ipopeopu	

MFP Retail

List of Measures		
ipcpbopc	ipcpslsu	ipopgmpv
ipcpbopu	ipcpbopr	ipoprecc
ipcpeopc	ipcpeopr	ipoprecu
ipcpeopu	ipcprecr	ipopslsr
ipcpgmpv	ipopbopc	ipopslsu
ipcprecc	ipopbopu	ipopbopr
ipcprecu	ipopeopc	ipopeopr
ipcpslsr	ipopeopu	ipoprecr

Example
loadactuals.ksh IP_MFP_transfer_list.txt 1

Error Information

Table 5–3   Load Actuals Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadactuals	40	yes	Arguments are missing.
loadactuals	41	yes	Domain path does not exist.
loadactuals	42	yes	Data file does not exist.
loadactuals	43	no	All measure input files are empty or missing.
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 the Classic Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Fusion Client* 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 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 IP

Script
exportToIP.ksh

Usage
exportToIP.ksh <maxprocesses> <destination> <measurelist>

Table 5–4   Export to IP 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 $MFP_HOME/bin directory. The file name should follow the convention of MFP_IP_transferList.txt.

Control File
MFP_IP_transfer_list.txt contains the following list of measures that can be exported as part of this script:

MFP Cost

List of Measures		
buopbopc buopbopc	buopeopu buopeopu	bucpeopc bucpeopc
buopeopc buopeopc	buoprecu buoprecu	bucpgmpv bucpgmpv
buopgmpv buopgmpv	buopslsu buopslsu	bucprecc bucprecc
buoprecc buoprecc	bucpbopu bucpbopu	bucpslsc bucpslsc
buopslsc buopslsc	bucpeopu bucpeopu	bucpslsnetr bucpslsnetr
buopslsnetr buopslsnetr	bucprecu bucprecu	bucpslsr bucpslsr
buopslsr buopslsr	bucpslsu bucpslsu	
buopbopu buopbopu	bucpbopc bucpbopc	

MFP Retail

List of Measures		
bucpbopc bucpbopc	buopbopc buopbopc	bucpbopu bucpbopu
bucpbopr bucpbopr	buopbopr buopbopr	bucpeopu bucpeopu
bucpeopc bucpeopc	buopeopc buopeopc	bucprecu bucprecu
bucpeopr bucpeopr	buopeopr buopeopr	bucpslsu bucpslsu
bucpgmpv bucpgmpv	buopgmpv buopgmpv	buopbopu buopbopu
bucpmkdr bucpmkdr	buopmkdr buopmkdr	buopeopu buopeopu
bucprecc bucprecc	buoprecc buoprecc	buoprecu buoprecu

List of Measures		
bucprecr bucprecr	buoprecr buoprecr	buopslsu buopslsu
bucpslsr buopbopc	buopslsr buopslsr	

Error Information

Table 5–5   Export to IP 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 IP are required.

- The script uses the RPAS exportMeasure utility to export measure data from the domain. 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* for details on this utility.

- The exportToIP.ksh script does not clear the export (approval) flags. In order to clear the export flags, the customer must run the script resetExportFlags.ksh. See Reset Export Flags for details.

Export to AP

Script
exportToAP.ksh

Usage
exportToAP.ksh <maxprocesses> <destination> <measurelist>

Table 5–6   Export to AP 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 `$MFP_HOME/bin` directory. The file name should follow the convention of MFP_AP_transferList.txt.
exportAtSubclas	Sets the product level for the exported measure data to be sub-class instead of class.	By default, the product level for the exported measure data is class. This argument changes the product level of the export to sub-class.

Control File

MFP_AP_transfer_list.txt contains the following list of measures that can be exported as part of this script:

MFP Cost

List of Measures		
bucpgmpv mfcpposgmr	bucpslsr mfcpslsr	bucpslsu mfcpslsu
bucprecc mfcprecc	bucpslsr mfcpslsr	

MFP Retail

List of Measures		
bucpgmpv mfcpposgmr	bucpslsr mfcpslsr	bucpslsu mfcpslsu
bucprecc mfcprecc	bucprecu mfcprecu	bucprecr mfcprecr

Error Information

Table 5–7   Export to AP 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 AP are required.

- The script uses the RPAS exportMeasure utility to export measure data from the domain. 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* for details on this utility.

- The exportToIP.ksh script does not clear the export (approval) flags. In order to clear the export flags, the customer must run the script resetExportFlags.ksh. See Reset Export Flags for details.

Reset Export Flags

Script

resetExportFlags.ksh

Usage

resetExportFlags.ksh

Table 5–8   Export Flag Measures

Measure Name	Measure Label
mowpappcpnewb	Wp Newly approved CP (used by ODI)

Table 5–8   (Cont.) Export Flag Measures

mowpappopnewb	Wp Newly approved OP (used by ODI)
modyappcpnewb	Wp Newly approved CP (used in export script)
modyappopnewb	Wp Newly approved OP (used in export script)

Notes

- This optional script is used to clear the Boolean measures listed in Table 5–8.

- If the customer will only be exporting MFP data to one external application (either Assortment Planning or Item Planning), the customer may wish to have these flags reset automatically after the export of the MFP data. To do so, the customer should uncomment the following two lines in exportToAP.ksh or in exportToIP.ksh:

```
# _call "$MFP_HOME/bin/resetExportFlags.ksh"
# _verify_script
```

  To uncomment these two lines, remove the "#" from the beginning of each line.

- If the customer will be exporting MFP data both to AP and to IP, the customer may wish to train the users to do the exports in a particular order and only uncomment the above two lines in one of the export scripts but leave them commented out in the other export script.

  For example, if the customer always exports MFP-to-AP followed by MFP-to-IP, then the customer could execute the script exportToAP.ksh as is (with the call to resetExportFlags.ksh commented out), which would leave the approval flags unchanged, but have exportToIP.ksh reset the flags by uncommenting out the above lines.

6

RMS and Retail Analytics Integration

This chapter describes the overall MFP data flow and integration with RMS, Oracle Retail Analytics (RA), markdown optimization applications, and data warehouses.

For information about integration for the Fashion Planning Bundle, see Chapter 4, "Fashion Planning Bundle Integration". For information about script integration for the bundle, see Chapter 5, "Script Integration".

Overview of the MFP Data Flow

Figure 6–1 shows the integration of MFP with other applications and the flow of data between the applications.

Figure 6–1 Data Flow for the Integration of MFP with Other Applications
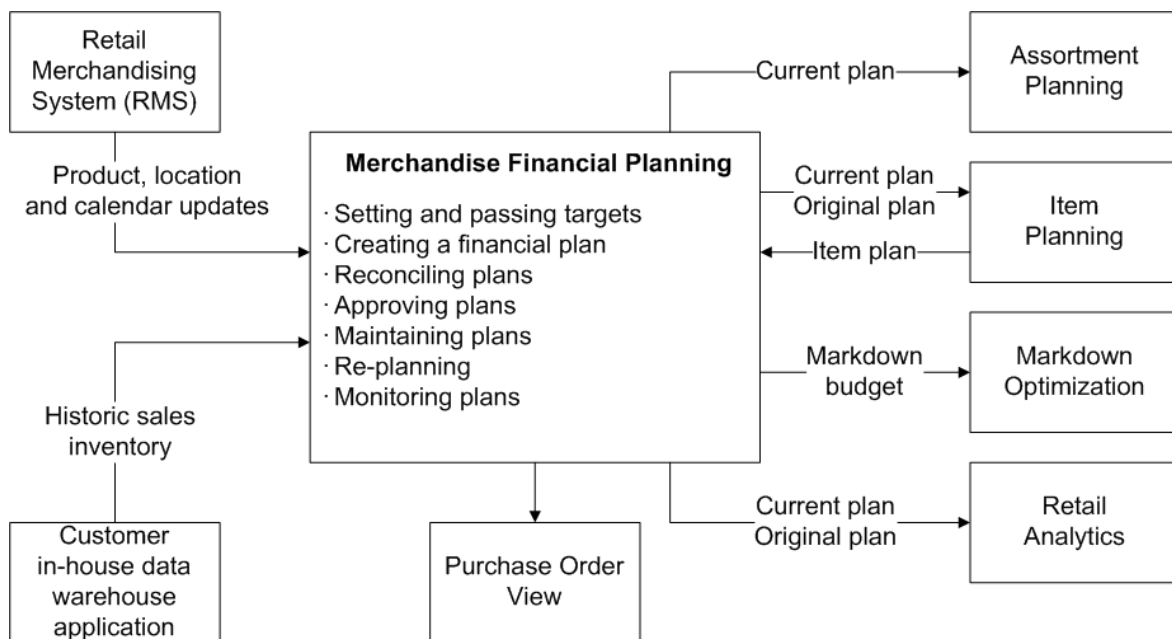
Integration Interface Data Flow Description

These descriptions explain some of the data flows in Figure 6–1. For information about the data flows among MFP, IP, and AP, see Chapter 4, "Fashion Planning Bundle Integration" and Chapter 5, "Script Integration".

From RMS to MFP

> **Note:** The integration between RMS and MFP includes only hierarchy, on order, and inventory data. All other data required by MFP are not part of the RMS/MFP integration.

Data for the following hierarchies is imported into MFP from RMS:

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

> **Note:** Non-stockholding company stores and non-inventory items which are non-merchandise items, consignment, concession and deposit returns are sent from RMS to MFP but are not utilized in MFP.

Hierarchies are the structures used by an organization to define the relationships that exist between measures of data, products, locations, time, and other dimensions. These dimensions are represented within the Fashion Planning Bundle applications as hierarchies that correspond to an organization's structure, including all roll-ups.

The Product hierarchy provides the parent-child merchandise level relationships that are available within an application. The Location hierarchy provides the parent-child-location level relationships that are available within an application. Application data is presented at an intersection level of the Product, Location, and Calendar hierarchies.

In addition to the hierarchy files, MFP receives on order and inventory from RMS. These files are based at the week/style-color/store level and then aggregated to the planning levels in the MFP domain.

Fore more information about the RMS integration, see RMS to MFP Transformation. For additional details on the RMS/MFP integration from the perspective of RMS, see the RPAS/MFP - RMS Integration chapter of the *RMS Operations Guide - Volume 1*.

From a Data Warehouse Application to MFP

The following data are imported into MFP from a customer in-house data warehouse application:

- Historic sales
- Inventory

> **Note:** There is no GA integration between Oracle Retail Merchandise Financial Planning and Oracle Retail Data Warehouse.

From MFP to a Markdown Optimization Engine

MFP Retail sends the current plan markdown budget to a markdown optimization application such as Oracle Retail Markdown Optimization.

For more information, see Markdown Optimization Integration.

From MFP to RA

The following data are sent from MFP to RA:

- Current Plan (Cp)

- Original Plan (Op)

RA uses these plans for detailed reporting.

RMS to MFP Transformation

This section details the process that is required to transform the extracted RMS files to create load ready files for the MFP solution.

Script Installation

1. The retlforRPAS directory and integration directory should be installed under the same directory. Both directory structures should overlap because MFP uses the schema and environment files of the retlforRPAS release. The retlforRPAS is part of the RMS release while the integration directory is part of the MFP release.

2. On some flavors of Unix, you may not have the nawk executable. You may only have the awk executable. In that case a soft link is needed to be created with the name nawk which will point to the awk executable.

3. RETL should be installed in the machine.

Environment Setup

The following environment variables need to be set in the environment:

- RMSE_RPAS_HOME points to the RMS release of RPAS integration scripts directory.

  ```
  export RMSE_RPAS_HOME=$MFP_HOME
  ```

- The DATA_DIR is the input and output directory. The RMS input files has to be copied to the DATA_DIR directory. After the MFP transformation, the output files also will be stored in the same location. Once the processing is done, the processed input files will be transfer to the processed directory in the DATA_DIR.

- RMSE_RPAS_HOME points to the RMS release of RPAS integration scripts directory.

  ```
  export RMSE_RPAS_HOME=$MFP_HOME
  ```

- As the RMS environment scripts use the RDF_HOME variable internally, this needs to be set.

  ```
  export RDF_HOME=$MFP_HOME
  ```

- The RMS Schema directory

  ```
  export RMS_RPAS_SCHEMA_DIR=$RMSE_RPAS_HOME/rfx/schema/
  ```

- The MFP Schema directory

  ```
  export MFP_SCHEMA_DIR=$MFP_HOME/rfx/schema/
  ```

- The intermediate data files are stored in the MFP_TEMP_DIR, If the input files are very huge, it is advisable to use a special Temp directory, rather than using the

default OS temp directory. By default /tmp is taken as MFP_TEMP_DIR if this variable is not set.

```
export MFP_TEMP_DIR=$MFP_HOME/data
```

In addition to the variable described above, the following optional environment variables can be set:

- ADD_AT_SIGN_TO_WH_DESC: This variable is used by the mfp_lochier.ksh. In the warehouse records if @ sign needs to be prefixed with warehouse name description field then this should be set to True. By default this feature is disabled. If this feature is required the following command should be executed in the environment.

```
export ADD_AT_SIGN_TO_WH_DESC=True
```

- PROCESS_WAREHOUSE_DATA: This variable is used by the mfp_lochier.ksh. By default this variable is set to True. If you do not want the mfp_lochier.ksh to process the warehouse data then this variable must be set as False.

```
export PROCESS_WAREHOUSE_DATA=False
```

- ENABLE_INVENTORY_RETAIL
  ENABLE_ON_ORDER_RETAIL

  These two variables are used by the mfp_inventory.ksh and mfp_on_order.ksh. The setting of these variables is dependent on the type of MFP being used. If you are using MFP Retail, you need to set these values to true to get the retail version of the inventory and on order files. By default both these variables are set to True. Use the following command in environment to disable them if you are instead using MFP Cost.

```
export ENABLE_INVENTORY_RETAIL=False
export
ENABLE_ON_ORDER_RETAIL=False
```

- ENABLE_INVENTORY_COST
  ENABLE_ON_ORDER_COST

  These two variables are used by the mfp_inventory.ksh and mfp_on_order.ksh. The setting of these variables is dependent on the type of MFP being used. If you are using MFP Cost, you need to set these values to true to get the cost version of the inventory and on order files. By default both these variables are set to True. Use the following command in environment to disable them if you are instead using MFP Retail.

```
export ENABLE_INVENTORY_COST=False
    export
ENABLE_ON_ORDER_COST=False
```

RMS Integration Script Details

This section outlines the scripts that will be run to transform the RMS files to load into MFP. These scripts reside in the $MFP_HOME/rfx/src directory. As previously stated, the input files and output files are in the $DATA_DIR.

- Script name: mfp_calhier.ksh
  Description: Transforms exported calendar hierarchy data from RMS to MFP-loadable format.
  Input data file(s): rmse_rpas_clndmstr.dat
  Output file(s): clnd.csv.dat
  Params: None

- Script name: mfp_prodhier.ksh
  Description: Transforms exported product hierarchy data from RMS to MFP-loadable format.
  Input file(s): rmse_rpas_item_master.dat, rmse_rpas_merchhier.dat
  Output file(s): prod.csv.dat
  Params: None

- Script name: mfp_lochier.ksh
  Description: Transforms exported location hierarchy data from RMS to MFP-loadable format.
  Input data file(s): rmse_rpas_orghier.dat, rmse_rpas_store.dat and rmse_rpas_wh.dat
  Output file(s): loc.csv.dat
  Params: None

- Script name: mfp_inventory.ksh
  Description: Transforms the extracted Inventory data from RMS to an MFP-loadable format.
  Required input data file: rmse_mfp_inventory.W.dat
  Optional Input data file: rmse_mfp_inventory.I.dat
  Output file(s):
  Params: None

- Script name: mfp_on_order.ksh
  Description: Transforms the extracted On-order data from RMS into an MFP-loadable format.
  Required input data file(s): rmse_mfp_onorder.dat
  Output file(s): mfp_inventory_retail.csv.ovr, mfp_inventory_cost.csv.ovr
  Params: None

Once the processing is done, the processed input files are transferred to the processed directory in the $DATA_DIR. The output files need to be transferred to the MFP domains input directory in order to load with the standard load scripts. For more about the standard load scripts, see Batch Designs.

Markdown Optimization Integration

The exportToMDO.ksh script is located in the MFP_HOME/bin folder. It is used to create MFP Retail current plan markdown budget extract for exports to markdown optimization applications such as Oracle Retail Markdown Optimization (MDO).

The script takes three arguments:

- The location for the destination directory.

- The RPAS name for a dimension along the location hierarchy. All location dimensions above or at the lowest stored dimension, channel (CHNL), are valid. You must provide a valid dimension.

- The RPAS name for a dimension along the product hierarchy. All product dimensions above or at the lowest stored dimension, subclass (SCLS), are valid. You must provide a valid dimension.

The script generates the total value of current approved markdowns. This is the sum of the promotional, permanent, and clearance markdowns at the base intersection. This is derived from the product and location dimensions that you supply.

> **Note:** The script assumes the calendar dimension to be month.

The markdowns, along with position names along the dimensions, are written to the ASH_BUDGET_TBL.dat file in the destination directory that you provide. It is a pipe delimited file that contains the following information:

- Product key

- Location key

- Fiscal year

- Fiscal month

- Total markdown

The product key (also known as the merchandise key) and location key are the position names for the product and location. The year is a four digit year. The month is an index of the fiscal month. For instance, if you start your fiscal year on February 1, the calendar month of December 2011 would be 2011|11.

RA Integration

Oracle Retail Analytics is a business intelligence solution for the retail industry. Retail Analytics offers an integrated view of retail data from various source systems, and it allows users to create analytical reports for areas such as merchandising and marketing, supply chain management, and corporate planning and performance management. Retail Analytics supports as-is, as-was, and point in time analysis methods and reporting.

Retail Analytics extracts data from the source systems and then transforms and loads the data into the Oracle Business Intelligence Repository to support reporting and analysis. MFP is an application from which Retail Analytics extracts data. Retail Analytics uses Oracle Data Integrator (ODI) for extraction, transformation, and loading. ODI programs that extract information from MFP are packaged with the Oracle Retail Analytics application. At the time of Retail Analytics installation, these ODI programs are deployed to the MFP instance. For more information about Oracle Retail Analytics and the ODI extraction programs, see the following guides:

- Oracle Retail Analytics Implementation Guide

- Oracle Retail Analytics Installation Guide

- Oracle Retail Analytics Operations Guide

Initial Setup

Before running either of the MFP to RA integration packages for the first time, domain support data must be initialized by running the following commands once per MFP domain:

```
mfp_integration_setup.ksh  -d  "{mfpcost-domain-path}"
mfp_integration_setup.ksh  -d  "{mfpretail-domain-path}"
```

This script generates mappings between the MFP Week positions and RA Week Number IDs. Similarly, it generates mappings between the MFP Channel positions and RA Channel Number IDs. Therefore, in an MFP domain, if Week positions are added to the Calendar hierarchy or if Channels are added to the Location hierarchy, then the mfp_integration_setup.ksh script must be run again to update the support data.

This script is located in the RA media pack.

Measure Data Integration

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

- SDE_MFPCostFact Package (MFP Cost to RA) is described in Data Mappings for SDE_MFPCostFact Package.

- SDE_MFPRetailFact Package (MFP Retail to RA) is described in Data Mappings for SDE_MFPRetailFact Package.

Scripts Used By the MFP to RA Packages

(None)

Data Mappings for SDE_MFPCostFact Package

The SDE_MFPCostFact package clears the data in the RA staging tables-W_RTL_MFPCPC_SC_CH_WK_FS and W_RTL_MFPOPC_SC_CH_WK_FS, and then populates data records wherever MFP's "Newly Approved" flag is "True" (mowpappcpnewb for Current Plan and mowpappopnewb for Original Plan).

Table 6–1   MFP Cost to RA Current Plan Cost (CPC) Data

MFP Cost Expression	W_RTL_MFPCPC_SC_CH_WK_FS Table
{Part of SCLS after last underscore.}	PROD_SC_NUM
{Part of CLSS after last underscore.}	PROD_CL_NUM

Table 6–1   (Cont.) MFP Cost to RA Current Plan Cost (CPC) Data

MFP Cost Expression	W_RTL_MFPCPC_SC_CH_WK_FS Table
DEPT	PROD_DP_NUM
RAWEEKNUM	MFP_WK_NUM
RACHANNELNUM	CHANNEL_NUM
BUCPSLSU	MFPCPC_SLS_QTY
BUCPSLSR	MFPCPC_SLS_RTL_AMT
BUCPSLSC	MFPCPC_SLS_COST_AMT
(BUCPSLSR / (1 + BUCPVATP)) * BUCPVATP	MFPCPC_TAX_RTL_AMT
BUCPSLSR / (1 + BUWPVATP)	MFPCPC_SLSTE_RTL_AMT
BUCPGMPV	MFPCPC_PROF_COST_AMT
BUCPBOPC	MFPCPC_BOH_COST_AMT
BUCPBOPU	MFPCPC_BOH_QTY
BUCPEOPC	MFPCPC_EOH_COST_AMT
BUCPEOPU	MFPCPC_EOH_QTY
BUCPRECC	MFPCPC_INVRC_COST_AMT
BUCPRECU	MFPCPC_INVRC_QTY
BUCPCHRINKC	MFPCPC_SHRINK_COST_AMT
BUCPCHRINKU	MFPCPC_SHRINK_QTY
BUCPMISCOUTC	MFPCPC_MISCO_COST_AMT
BUCPMISCOUTU	MFPCPC_MISCO_QTY
BUCPMISCINC	MFPCPC_MISCI_COST_AMT
BUCPMISCINU	MFPCPC_MISCI_QTY
BUCPDEVALC	MFPCPC_DVAL_COST_AMT
RASOURCENUM	DATASOURCE_NUM_ID
RACURRENCYCODE	DOC_CURR_CODE
{Part of SCLS after last underscore.} || '~' || RAWEEKNUM || '~' || RACHANNELNUM	INTEGRATION_ID

Table 6–2   MFP Cost to RA Original Plan Cost (OPC) Data

MFP Cost Expression	W_RTL_MFPOPC_SC_CH_WK_FS Table
{Part of SCLS after last underscore.}	PROD_SC_NUM
{Part of CLSS after last underscore.}	PROD_CL_NUM
DEPT	PROD_DP_NUM
RAWEEKNUM	MFP_WK_NUM
RACHANNELNUM	CHANNEL_NUM
BUCPSLSU	MFPOPC_SLS_QTY

Table 6–2   (Cont.) MFP Cost to RA Original Plan Cost (OPC) Data

MFP Cost Expression	W_RTL_MFPOPC_SC_CH_WK_FS Table
BUCPSLSR	MFPOPC_SLS_RTL_AMT
BUCPSLSC	MFPOPC_SLS_COST_AMT
(BUCPSLSR / (1 + BUCPVATP)) * BUCPVATP	MFPOPC_TAX_RTL_AMT
BUCPSLSR / (1 + BUWPVATP)	MFPOPC_SLSTE_RTL_AMT
BUCPGMPV	MFPOPC_PROF_COST_AMT
BUCPBOPC	MFPOPC_BOH_COST_AMT
BUCPBOPU	MFPOPC_BOH_QTY
BUCPEOPC	MFPOPC_EOH_COST_AMT
BUCPEOPU	MFPOPC_EOH_QTY
BUCPRECC	MFPOPC_INVRC_COST_AMT
BUCPRECU	MFPOPC_INVRC_QTY
BUCPCHRINKC	MFPOPC_SHRINK_COST_AMT
BUCPCHRINKU	MFPOPC_SHRINK_QTY
BUCPMISCOUTC	MFPOPC_MISCO_COST_AMT
BUCPMISCOUTU	MFPOPC_MISCO_QTY
BUCPMISCINC	MFPOPC_MISCI_COST_AMT
BUCPMISCINU	MFPOPC_MISCI_QTY
BUCPDEVALC	MFPOPC_DVAL_COST_AMT
RASOURCENUM	DATASOURCE_NUM_ID
RACURRENCYCODE	DOC_CURR_CODE
{Part of SCLS after last underscore.} || '~' || RAWEEKNUM || '~' || RACHANNELNUM	INTEGRATION_ID

Data Mappings for SDE_MFPRetailFact Package

The SDE_MFPRetailFact package clears the data in the RA staging tables W_RTL_MFPCPR_SC_CH_WK_FS and W_RTL_MFPOPR_SC_CH_WK_FS, and then populates data records wherever MFP's "Newly Approved" flag is "True" (mowpappcpnewb for Current Plan and mowpappopnewb for Original Plan).

Table 6–3   MFP Retail to RA Current Plan Retail (CPR) Data

MFP Retail Expression	W_RTL_MFPCPR_SC_CH_WK_FS Table
{Part of SCLS after last underscore.}	PROD_SC_NUM
{Part of CLSS after last underscore.}	PROD_CL_NUM
DEPT	PROD_DP_NUM
RAWEEKNUM	MFP_WK_NUM
RACHANNELNUM	CHANNEL_NUM
BUCPSLSREGR	MFPCPR_SLSRG_RTL_AMT

Table 6–3   (Cont.) MFP Retail to RA Current Plan Retail (CPR) Data

MFP Retail Expression	W_RTL_MFPCPR_SC_CH_WK_FS Table
BUCPSLSPROR	MFPCPR_SLSPR_RTL_AMT
BUCPSLSCLRR	MFPCPR_SLSCL_RTL_AMT
BUCPSLSREGU	MFPCPR_SLSRG_QTY
BUCPSLSPROU	MFPCPR_SLSPR_QTY
BUCPSLSCLRU	MFPCPR_SLSCL_QTY
BUCPVATR	MFPCPR_TAX_RTL_AMT
(BUCPSLSREGR + BUCPSLSCLRR + BUCPSLSPROR) / (1 + BUWPVATP)	MFPCPR_SLSTE_RTL_AMT
BUCPIGMPV	MFPCPR_MARGIN_RTL_AMT
BUCPICOGSC	MFPCPR_COGS_COST_AMT
BUCPMKDPERMR	MFPCPR_MKDNPM_RTL_AMT
BUCPMKDPROMOR	MFPCPR_MKDNPR_RTL_AMT
BUCPMKDCLRR	MFPCPR_MKDNCL_RTL_AMT
BUCPMKUPR	MFPCPR_MKUP_RTL_AMT
BUCPBOPC	MFPCPR_BOH_COST_AMT
BUCPBOPR	MFPCPR_BOH_RTL_AMT
BUCPBOPU	MFPCPR_BOH_QTY
BUCPEOPC	MFPCPR_EOH_COST_AMT
BUCPEOPR	MFPCPR_EOH_RTL_AMT
BUCPEOPU	MFPCPR_EOH_QTY
BUCPRECC	MFPCPR_INVRC_COST_AMT
BUCPRECR	MFPCPR_INVRC_RTL_AMT
BUCPRECU	MFPCPR_INVRC_QTY
BUCPRHRINKR	MFPCPR_SHRINK_RTL_AMT
BUCPRHRINKU	MFPCPR_SHRINK_QTY
BUCPMISCOUTR	MFPCPR_MISCO_RTL_AMT
BUCPMISCOUTU	MFPCPR_MISCO_QTY
BUCPMISCINR	MFPCPR_MISCI_RTL_AMT
BUCPMISCINU	MFPCPR_MISCI_QTY
RASOURCENUM	DATASOURCE_NUM_ID
RACURRENCYCODE	DOC_CURR_CODE
{Part of SCLS after last underscore.} || '~' || RAWEEKNUM || '~' || RACHANNELNUM	INTEGRATION_ID

Table 6–4   MFP Retail to RA Original Plan Retail (OPR) Data

MFP Retail Expression	W_RTL_MFPOPR_SC_CH_WK_FS Table
{Part of SCLS after last underscore.}	PROD_SC_NUM
{Part of CLSS after last underscore.}	PROD_CL_NUM
DEPT	PROD_DP_NUM
RAWEEKNUM	MFP_WK_NUM
RACHANNELNUM	CHANNEL_NUM
BUCPSLSREGR	MFPOPR_SLSRG_RTL_AMT
BUCPSLSPROR	MFPOPR_SLSPR_RTL_AMT
BUCPSLSCLRR	MFPOPR_SLSCL_RTL_AMT
BUCPSLSREGU	MFPOPR_SLSRG_QTY
BUCPSLSPROU	MFPOPR_SLSPR_QTY
BUCPSLSCLRU	MFPOPR_SLSCL_QTY
BUCPVATR	MFPOPR_TAX_RTL_AMT
(BUCPSLSREGR + BUCPSLSCLRR + BUCPSLSPROR) / (1 + BUWPVATP)	MFPOPR_SLSTE_RTL_AMT
BUCPIGMPV	MFPOPR_MARGIN_RTL_AMT
BUCPICOGSC	MFPOPR_COGS_COST_AMT
BUCPMKDPERMR	MFPOPR_MKDNPM_RTL_AMT
BUCPMKDPROMOR	MFPOPR_MKDNPR_RTL_AMT
BUCPMKDCLRR	MFPOPR_MKDNCL_RTL_AMT
BUCPMKUPR	MFPOPR_MKUP_RTL_AMT
BUCPBOPC	MFPOPR_BOH_COST_AMT
BUCPBOPR	MFPOPR_BOH_RTL_AMT
BUCPBOPU	MFPOPR_BOH_QTY
BUCPEOPC	MFPOPR_EOH_COST_AMT
BUCPEOPR	MFPOPR_EOH_RTL_AMT
BUCPEOPU	MFPOPR_EOH_QTY
BUCPRECC	MFPOPR_INVRC_COST_AMT
BUCPRECR	MFPOPR_INVRC_RTL_AMT
BUCPRECU	MFPOPR_INVRC_QTY
BUCPRHRINKR	MFPOPR_SHRINK_RTL_AMT
BUCPRHRINKU	MFPOPR_SHRINK_QTY
BUCPMISCOUTR	MFPOPR_MISCO_RTL_AMT
BUCPMISCOUTU	MFPOPR_MISCO_QTY
BUCPMISCINR	MFPOPR_MISCI_RTL_AMT
BUCPMISCINU	MFPOPR_MISCI_QTY
RASOURCENUM	DATASOURCE_NUM_ID
RACURRENCYCODE	DOC_CURR_CODE

Table 6–4 (Cont.) MFP Retail to RA Original Plan Retail (OPR) Data

MFP Retail Expression	W_RTL_MFPOPR_SC_CH_WK_FS Table
{Part of SCLS after last underscore.} || '~' || RAWEEKNUM || '~' || RACHANNELNUM	INTEGRATION_ID

Configuring RPAS JDBC for ODI to Facilitate Retail Analytics Integration

These configuration instructions are relevant only for the integration between Retail Analytics and Merchandise Financial Planning (MFP). F or more information about Retail Analytics and its use of ODI, see the Oracle Retail Analytics documentation set.

Customers who use RPAS JDBC with Oracle Data Integrator (ODI) must perform the following steps:

1. Install the RPAS JDBC drivers on the ODI server:

    a. Extract the file jdbcclient.tar.gz to create the jdbcclient directory:

    ```
    cd "$RPAS_HOME"
    unzip -q jdbcclient.tar.zip
    tar xf jdbcclient.tar
    ```

    b. Copy the following four files to the ODI server under the directory $ODI_HOME/drivers:

    ```
    jdbcclient/driver/lib/ORjc.jar
    jdbcclient/driver/lib/ORssl14.jar
    jdbcclient/driver/lib/iaik_jce_full.jar
    jdbcclient/spy/lib/ORy.jar
    ```

    c. If the jdbcclient directory is on the same machine as the ODI server, execute the following commands:

    ```
    cd  "$RPAS_HOME/jdbcclient/driver/lib"
    cp -p ORjc.jar  ORssl14.jar  iaik_jce_full.jar  "$ODI_HOME/drivers"
    cp -p ../../spy/lib/ORy.jar  "$ODI_HOME/drivers"
    ```

2. From $ODI_HOME/bin directory, stop and restart the ODI agent.

    ```
    agentstop.sh -port=xxxx
    agent.sh -port=1055 -name=<agent_name>
    ```

    > **Note:** The startup scripts are set to automatically add all *.jar files in the ODI_HOME/drivers directory to the agent CLASSPATH.

3. Retrieve the TECH_RPASJDBC.xml file from $RPAS_HOME/ODI directory and import the file to ODI work repository. For importing instructions, see the *Oracle Retail Analytics Installation Guide*.

Configuring PO View

POView is designed around Database views which depend on various tables in RMS. The application pulls Open and Pending Purchase Orders from RMS. This section covers the following topics:

- Installation and Setup of Fusion Client for MFP with POView
- Overview of the PO View Dataflow

Installation and Setup of Fusion Client for MFP with POView

To install and set up the Fusion Client for MFP to use POView:

1. Install Oracle Retail Extract, Transform, and Load (RETL) for RPAS.

2. Transform the hierarchy, on-order, and inventory extracts from RMS to MFP readable format.

   There are separate scripts, each for product hierarchy, calendar, location, inventory data, and on-order data. The scripts are available in the MFP CD under MFP_HOME/rfx/src. For example, after running mfp_calhier.ksh, the rmse_rpas_clndmstr.dat file is transformed into MFP readable format in the clnd.csv.dat file.

3. Build the MFP Cost and Retail domains with the transformed hierarchy files.

4. Load the hierarchy files into the MFP Cost and Retail domains using the loadHier command.

5. Load On-order and Inventory files into MFP Cost and Retail by running the loadmeasure utility.

6. Install Fusion Client using the installer and connect to the MFP domain. Detailed steps for installing Fusion Client are available in the *Oracle Retail Predictive Application Server Installation Guide*.

7. Install POView using the installer. The database objects needed for POView will be created during installation or manually. In case the datasource is not setup/available, users can manually create the objects post install. Detailed steps for installing POView are available in the *Oracle Retail Predictive Application Server Installation Guide*. See the Installing Solution Plug-ins section.

8. Reinstall Fusion Client with Solution Plugin set to POView.

For more details on setup and configuration of POView, refer to the *Oracle Retail Predictive Application Server Configuration Tools User Guide* or *Oracle Retail Predictive Application Server Installation Guide*.

Overview of the PO View Dataflow

The following figure shows the interaction between RMS and POView.

Figure 6–2   Relationship between RMS and POView

The following table shows the list of RMS tables used by POView:

Table 6–5   Tables used by POView

DEPS
CLASS
SUBCLASS
ITEM_MASTER
COMPHEAD
CHAIN
STORE
WH
STORE_HIERARCHY
CHANNELS
ORDHEAD
ORDLOC

The following Database Objects need to be created in RMS:

> **Note:**   RMS uses a relational database, hence the SQL statements.

Example 6–1   SQL to create Custom Data Types

```
CREATE OR REPLACE TYPE POVIEW_DEPARTMENTIDS AS TABLE OF NUMBER(10);
CREATE OR REPLACE TYPE POVIEW_CLASS1IDS AS TABLE OF NUMBER(10);
CREATE OR REPLACE TYPE POVIEW_SUBCLASSIDS AS TABLE OF NUMBER(10);
CREATE OR REPLACE TYPE POVIEW_CHAINIDS AS TABLE OF NUMBER(10);
CREATE OR REPLACE TYPE POVIEW_CHANNELIDS AS TABLE OF NUMBER(10);
CREATE OR REPLACE TYPE POVIEW_AREAIDS AS TABLE OF NUMBER(10);
CREATE OR REPLACE TYPE POVIEW_DISTRICTIDS AS TABLE OF NUMBER(10);
CREATE OR REPLACE TYPE POVIEW_REGIONIDS AS TABLE OF NUMBER(10);
CREATE OR REPLACE TYPE POVIEW_LOCATIONIDS AS TABLE OF NUMBER(10);
CREATE OR REPLACE TYPE POVIEW_SUBCLASSCLASS1IDS AS TABLE OF NUMBER(10);
CREATE OR REPLACE TYPE POVIEW_SUBCLASSDEPARTMENTIDS AS TABLE OF NUMBER(10);
CREATE OR REPLACE TYPE POVIEW_CLASS1DEPARTMENTIDS AS TABLE OF NUMBER(10);
```

> **Note:** When a user launches PO View, they have selected a list of departments, classes, channels, etc. The corresponding IDs are temporarily held in the placeholders created by the above SQL. This information is then used to populate the views below that are used to hold the information required for the PO View dialog box.

Example 6–2   SQL for POVIEW_ITEM_MASTER_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_ITEM_MASTER_VIEW" ("ITEM", "ITEM_DESC",
"DEPT", "DEPT_NAME", "CLASS", "CLASS_NAME", "SUBCLASS", "SUB_NAME", "STAND-ARD_
UOM") AS
  Select Im.Item,im.item_desc, Im.Dept,Dp.Dept_Name, Im.Class,Cl.Class_Name,
Im.Subclass,Sc.Sub_Name, Im.Standard_Uom
  From Item_Master Im, Deps Dp, Class Cl, Subclass Sc
  Where Im.Dept = Dp.Dept And Im.Dept = cl.dept and IM.CLASS = cl.class and
Im.Dept = sc.dept and im.class = sc.class and
  im.subclass = sc.subclass;
```

Example 6–3   SQL for POVIEW_ORDHEAD_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_ORDHEAD_VIEW" ("NOT_AFTER_DATE", "NOT_BEFORE_
DATE", "OTB_EOW_DATE", "EARLIEST_SHIP_DATE", "LATEST_SHIP_DATE", "COMMENT_DESC",
"SUPPLIER", "CURRENCY_CODE", "ORDER_NO", "STATUS") AS
  Select Oh.Not_After_Date, Oh.Not_Before_Date, Oh.Otb_Eow_Date,
    Oh.Earliest_Ship_Date, Oh.Latest_Ship_Date, Oh.Comment_Desc, Oh.Supplier,
    oh.currency_code, oh.order_no, oh.status from ordhead oh;
```

Example 6–4   SQL for POVIEW_ORDLOC_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_ORDLOC_VIEW" ("ORDER_NO", "QTY_ORDERED",
"QTY_RECEIVED", "QTY_CANCELLED", "ESTIMATED_INSTOCK_DATE", "ITEM", "LOCATION",
"LOC_TYPE", "UNIT_COST", "UNIT_RETAIL") AS
  Select Ol.Order_No, Ol.Qty_Ordered, Ol.Qty_Received, Ol.Qty_Cancelled,
    Ol.Estimated_Instock_Date, Ol.Item, Ol.Location, Ol.Loc_Type,
    ol.unit_cost, ol.unit_retail from ordloc ol;
```

Example 6–5   SQL for POVIEW_STORE_HIERARCHY_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_STORE_HIERARCHY_VIEW" ("COMPANY", "CHAIN",
"AREA", "DISTRICT", "REGION", "STORE") AS
  Select SH.COMPANY, Sh.Chain, Sh.Area, Sh.District, sh.region, SH.STORE from
STORE_HIERARCHY SH;
```

Example 6–6   SQL for POVIEW_STORE_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_STORE_VIEW" ("STORE", "CHANNEL_ID",
"CURRENCY_CODE", "STORE_NAME10") AS
 Select S.STORE, S.CHANNEL_ID, S.CURRENCY_CODE, S.STORE_NAME10  From STORE S;
```

Example 6–7   SQL for POVIEW_SUPS_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_SUPS_VIEW" ("SUPPLIER", "SUP_NAME") AS
  Select sp.supplier, sp.sup_name from sups sp;
```

Example 6–8   SQL for POVIEW_WH_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_WH_VIEW" ("WH", "WH_NAME", "CHANNEL_ID",
"ORG_HIER_TYPE", "ORG_HIER_VALUE", "CURRENCY_CODE") AS
  Select wh.wh, wh.WH_NAME, WH.CHANNEL_ID, WH.ORG_HIER_TYPE, WH.ORG_HIER_VALUE,
WH.CURRENCY_CODE from wh wh;
```

Example 6–9 SQL for POVIEW_AREA_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_AREA_VIEW" ("AREA_NAME", "AREA") AS
  Select Area_Name, Area From Area;
```

Example 6–10 SQL for POVIEW_CHAIN_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_CHAIN_VIEW" ("CHAIN_NAME", "CHAIN") AS
  Select Chain_Name, Chain From CHAIN;
```

Example 6–11 SQL for POVIEW_CHANNELS_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_CHANNELS_VIEW" ("CHANNEL_NAME", "CHANNEL_ID")
AS
  Select Channel_Name, Channel_Id From Channels;
```

Example 6–12 SQL for POVIEW_COMPHEAD_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_COMPHEAD_VIEW" ("CO_NAME", "COMPANY") AS
  Select Co_Name, Company From Comphead;
```

Example 6–13 SQL for POVIEW_DEPS_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_DEPS_VIEW" ("DEPT_NAME", "DEPT") AS
  Select Dept_Name, Dept From Deps;
```

Example 6–14 SQL for POVIEW_DISTRICT_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_DISTRICT_VIEW" ("DISTRICT_NAME", "DISTRICT")
AS
  Select District_Name, District From District;
```

Example 6–15 SQL for POVIEW_REGION_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_DISTRICT_VIEW" ("DISTRICT_NAME", "DISTRICT")
AS
  Select District_Name, District From District;
```

Example 6–16 SQL for POVIEW_CLASS_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_CLASS_VIEW" ("CLASS_NAME", "CLASS", "DEPT")
AS
  Select Class_Name, Class, Dept From Class;
```

Example 6–17 SQL for POVIEW_SUBCLASS_VIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW_SUBCLASS_VIEW" ("SUB_NAME", "SUBCLASS",
"DEPT", "CLASS") AS
  Select Sub_Name, Subclass, Dept, Class From Subclass;
```

> **Note:** The above set of views (6 - 2 to 6 - 17) collect the building
> blocks of the data required to populate the PO View dialog box. The
> view below (6 - 18) collates all the data together into the form that will
> be displayed in the PO View dialog box.

Example 6–18 SQL for POVIEW

```
CREATE OR REPLACE FORCE VIEW "POVIEW" ("ORDER_NO", "NOT_AFTER_DATE", "NOT_BEFORE_
DATE", "OTB_EOW_DATE", "EARLIEST_SHIP_DATE", "LATEST_SHIP_DATE", "QTY_ORDERED",
 "QTY_RECEIVED", "QTY_CANCELLED", "ESTIMATED_INSTOCK_DATE", "COMMENT_DESC",
"ITEM", "LOC_TYPE", "SUPPLIER", "SUP_NAME", "ORDER_COST_CURRENCY_CODE", "UNIT_
```

```
COST", "UNIT_RETAIL", "ITEM_DESC", "DEPARTMENT", "DEPT_NAME", "CLASS1", "CLASS_
NAME", "SUBCLASS", "SUB_NAME", "STANDARD_UOM", "LOCATION", "LOC_NAME", "COMPANY",
"CHAIN", "AREA", "DISTRICT", "REGION", "CHANNEL", "ORG_HIER_TYPE", "ORG_HIER_
VALUE", "RETAIL_CURRENCY_CODE") AS

  SELECT OL.ORDER_NO,
  OH.NOT_AFTER_DATE,
  OH.NOT_BEFORE_DATE,
  OH.OTB_EOW_DATE,
  OH.EARLIEST_SHIP_DATE,
  OH.LATEST_SHIP_DATE,
  OL.QTY_ORDERED,
  OL.QTY_RECEIVED,
  OL.QTY_CANCELLED,
  OL.ESTIMATED_INSTOCK_DATE,
  OH.COMMENT_DESC,
  OL.ITEM,
  OL.LOC_TYPE,
  OH.SUPPLIER,
  SP.SUP_NAME,
  OH.CURRENCY_CODE ORDER_COST_CURRENCY_CODE,
  OL.UNIT_COST,
  Ol.Unit_Retail,
  im.item_desc,
  Im.Dept Department,
  IM.DEPT_NAME,
  Im.Class Class1,
  IM.CLASS_NAME,
  Im.Subclass,
  IM.SUB_NAME,
  IM.STANDARD_UOM,
  S.STORE LOCATION,
  S.STORE_NAME10 LOC_NAME,
  SH.COMPANY COMPANY,
  Sh.Chain Chain,
  Sh.Area Area,
  Sh.District District,
  sh.region region,
  S.CHANNEL_ID Channel,
  NULL ORG_HIER_TYPE,
  NULL ORG_HIER_VALUE,
  S.CURRENCY_CODE RETAIL_CURRENCY_CODE
FROM Poview_Ordloc_View OL,
  Poview_Ordhead_View OH,
  Poview_Sups_View SP,
  Poview_Item_Master_View IM,
  Poview_Store_View S,
  Poview_Store_Hierarchy_View SH
WHERE OH.STATUS = 'A'
AND OL.QTY_ORDERED - (OL.QTY_RECEIVED + NVL(OL.QTY_CANCELLED, 0)) > 0
AND OL.LOC_TYPE = 'S'
AND OL.ORDER_NO = OH.ORDER_NO
AND OH.SUPPLIER = SP.SUPPLIER
AND OL.ITEM     = IM.ITEM
AND OL.LOCATION = S.STORE
AND S.STORE     = SH.STORE
UNION ALL
SELECT OL.ORDER_NO,
  OH.NOT_AFTER_DATE,
  OH.NOT_BEFORE_DATE,
```

```
            OH.OTB_EOW_DATE,
            OH.EARLIEST_SHIP_DATE,
            OH.LATEST_SHIP_DATE,
            OL.QTY_ORDERED,
            OL.QTY_RECEIVED,
            OL.QTY_CANCELLED,
            OL.ESTIMATED_INSTOCK_DATE,
            OH.COMMENT_DESC,
            OL.ITEM,
            OL.LOC_TYPE,
            OH.SUPPLIER,
            SP.SUP_NAME,
            OH.CURRENCY_CODE ORDER_COST_CURRENCY_CODE,
            OL.UNIT_COST,
            Ol.Unit_Retail,
            im.item_desc,
            Im.Dept Department,
            IM.DEPT_NAME,
            Im.Class Class1,
            IM.CLASS_NAME,
            Im.Subclass,
            IM.SUB_NAME,
            IM.STANDARD_UOM,
            WH.WH LOCATION,
            WH.WH_NAME LOC_NAME,
            NULL COMPANY,
            Null Chain,
            Null Area,
            Null District,
            Null region,
            WH.CHANNEL_ID Channel,
            WH.ORG_HIER_TYPE ORG_HIER_TYPE,
            WH.ORG_HIER_VALUE ORG_HIER_VALUE,
            WH.CURRENCY_CODE RETAIL_CURRENCY_CODE
        FROM Poview_Ordloc_View OL,
          Poview_Ordhead_View OH,
          Poview_Sups_View SP,
          Poview_Item_Master_View IM,
          Poview_Wh_View WH
        WHERE OH.STATUS = 'A'
        AND OL.QTY_ORDERED - (OL.QTY_RECEIVED + NVL(OL.QTY_CANCELLED, 0)) > 0
        AND OL.LOC_TYPE = 'W'
        AND OL.ORDER_NO = OH.ORDER_NO
        AND OH.SUPPLIER = SP.SUPPLIER
        And Ol.Item    = Im.Item
        And Ol.Location = Wh.Wh;
```

7

Batch Processing

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

Batch Structure Overview

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

Table 7–1   Directories Used by Batch Scripts

Directory Name	Content of the Directory
Bin	Batch scripts
Config	MFP 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 `$MFP_HOME/bin` directory where the batch script is found.
- Dependencies on other batch scripts.

For a detailed description of each script, see Batch Designs.

Daily Batch Scripts

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

Table 7–2 lists information on the daily batch script. For more information on the Load Actuals and Export to IP scripts, see Integration Scripts in Chapter 5.

Table 7–2   Daily Batch Scripts

Description	Script Name	Batch Directory	Dependency
Backup	N/A	N/A	N/A
Export to AP	exportToAP.ksh	bin	Backup
Export to IP	exportToIP.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 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 AP	exportToAP.ksh	bin	Backup
Export to IP	exportToIP.ksh	bin	Backup
Calendar Hierarchy Load	loadhier.ksh	loadhier	Backup
Product Hierarchy Load	loadhier.ksh	loadhier	Backup
Location Hierarchy Load	loadhier.ksh	loadhier	Backup
Load on order 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 IP data	loadactuals.ksh	bin	Calendar Hierarchy Load, Product Hierarchy Load, Location Hierarchy Load, and export file ready from IP
Propagate inventory and aggregate data for all planning levels	processactuals.ksh	actualize	Load on order data, Load Actuals data
Generate sales forecast	runforecast.ksh	forecast	Load on order data, Load Actuals data
Refresh existing workbooks **Note:** This script is optional.	refresh.ksh	workbook	Generate sales forecast

Table 7–3   (Cont.)   Weekly Batch Scripts

Description	Script Name	Batch Directory	Dependency
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 as a one-off process 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

The environment.ksh script is sourced by every batch script in MFP. This script sets the following environment variables for use by the other batch scripts:

- export MFP_CONFIGNAME=mfprtl for MFP Retail and mfpcst for MFP Cost

- export MFP_DOMAINHOME=$MFP_HOME/domain

- export MFP_MASTERDOMAIN=$MFP_DOMAINHOME/mfprtl for MFP Retail and mfpcst for MFP Cost

- export MFP_CONFIGHOME=$MFP_HOME/config

- export MFP_EXPORT=$MFP_HOME/export

- export MFP_INPUTHOME=$MFP_HOME/input

- export MFP _LOG_DIR=$MFP_HOME/logs

- export MFP _LIB=$MFP_HOME/bin

- export MFP _TEMP=$MFP_HOME/temp

- export MFP _BATCH=$MFP_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 IP 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.

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

Table 7–6 Calendar Hierarchy Load Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadhier	3	yes	Hierarchy file does not exist.
loadhier	6	yes	One or more arguments are missing.
loadhier	13	yes	Domain does not exist.
loadhier	Other non-zero value	yes	Loading the hierarchy produced errors. Further batch processing should not be run until the error has been investigated.

Notes

- Before running this script, a clnd.dat file must exist in the input directory of the domain. This can be created by running the transformation scripts to transform the RMS files. For more information, see Chapter 6, "RMS and Retail Analytics Integration".

- This script uses the RPAS loadHier utility. 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* 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 7–7   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

Table 7–8   Product Hierarchy Load Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadhier	3	yes	Hierarchy file does not exist.
loadhier	6	yes	One or more arguments are missing.
loadhier	13	yes	Domain does not exist.
loadhier	Other non-zero value	yes	Loading the hierarchy produced errors. Further batch processing should not be run until the error has been investigated.

Notes

- Before running this script, a prod.dat file must exist in the input directory of the domain. This can be created by running the transformation scripts to transform the RMS files. For more information, see Chapter 6, "RMS and Retail Analytics Integration".

- This script uses the RPAS loadHier utility. 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* 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 7–9   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.
purgeage	Sets the number of days after updating a hierarchy position before purging the position.	The default is 10000.

Example
loadhier.ksh loc 10000

Table 7–10   Location Hierarchy Load Error Information

Task Name	Error Code	Abort Required?	Description of Error
loadhier	3	yes	Hierarchy file does not exist.
loadhier	6	yes	One or more arguments are missing.
loadhier	13	yes	Domain does not exist.
loadhier	Other non-zero value	yes	Loading the hierarchy produced errors. Further batch processing should not be run until the error has been investigated.

Notes

- Before running this script, a loc.dat file must exist in the input directory of the domain. This can be created by running the transformation scripts to transform the RMS files. For more information, see Chapter 6, "RMS and Retail Analytics Integration".

- This script uses the RPAS loadHier utility. 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* for details on this utility.

- The calendar, product, and location hierarchy loads can be run in any order.

Load On Order Data

Script
loadactuals.ksh

Usage
loadactuals.ksh <measurelist> <maxprocesses>

Table 7–11   Load On Order Data

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 `$MFP_HOME/bin` directory. The file name should follow the convention of IP_MFP_transferList.txt.
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.

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

MFP Cost

- drtyonordc
- drtyonordu

MFP Retail

- drtyonordc
- drtyonordu
- drtyonordr

Example
loadactuals.ksh LoadOnOrderList.txt 1

Error Information

Table 7–12   Load Actuals Error Information

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

Notes

- If you are using RMS on order file, ensure that you run the transformation and transfer the load ready file to the input directory of the domain. For more information, see Chapter 6, "RMS and Retail Analytics Integration".

- This script uses the RPAS loadmeasure utility. See the *RPAS Administration Guide for the Classic Client* or the *RPAS Administration Guide for the Fusion Client* 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 file name.

- The script does not produce an error when records are rejected from the loaded files. These rejected records are logged in the log output of the load process.

Load Actuals Data

Script
loadactuals.ksh

Usage
loadactuals.ksh <measurelist> <maxprocesses>

Table 7–13   Load Actuals Data

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 $MFP_HOME/bin directory. The file name should follow the convention of IP_MFP_transferList.txt.
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.

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

MFP Cost

drtybopclrc	drtybopclru	drtybopregc
drtybopregu	drtyeopclrc	drtyeopclru
drtyeopregc	drtyeopregu	drtyrecc
drtyrecu	drtyslsclrr	drtyslsclru
drtyslsclrc	drtyslspror	drtyslsprou
drtyslsproc	drtyslsregr	drtyslsregu
drtyslsregc	drtydevalc	buwpvatp
drtyshrinkc	drtyshrinku	drtymiscinc
drtymiscinu	drtymiscoutc	drtymiscoutu

MFP Retail

drtybopclrc	drtybopclru	drtybopclrr
drtybopregc	drtybopregu	drtybopregr
drtyeopclrc	drtyeopclru	drtyeopclrr
drtyeopregc	drtyeopregu	drtyeopregr
drtyrecc	drtyrecu	drtyrecr
drtyslsclrr	drtyslsclru	drtyslspror
drtyslsprou	drtyslsregr	drtyslsregu
buwpvatp	drtyshrinkr	drtyshrinku
drtymiscinr	drtymiscinu	drtymiscoutr
drtymiscoutu	drtymkdr	drtymkupr

Example
loadactuals.ksh LoadOnOrderList.txt 1

Error Information

Table 7–14   Load Actuals Data Error Information

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

Notes

- If you are using RMS inventory file, ensure that you run the transformation and transfer the load ready file to the input directory of the domain. For more information, see Chapter 6, "RMS and Retail Analytics Integration".

- This script uses the RPAS loadmeasure utility. 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* for details.

- 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 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 IP Data

Script
loadactuals.ksh

Usage
loadactuals.ksh <measurelist> <maxprocesses>

Table 7–15   Load IP Data

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 `$MFP_HOME/bin` directory. The file name should follow the convention of IP_MFP_transferList.txt.
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.

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

MFP Cost

ipcpbopc ipcpbopc	ipcpbopu ipcpbopu	ipcpeopc ipcpeopc
ipcpeopu ipcpeopu	ipcpgmpv ipcpgmpv	ipcprecc ipcprecc
ipcprecu ipcprecu	ipcpslsc ipcpslsc	ipcpslsr ipcpslsr
ipcpslsu ipcpslsu	ipopbopc ipopbopc	ipopbopu ipopbopu
ipopeopc ipopeopc	ipopeopu ipopeopu	ipopgmpv ipopgmpv
ipoprecc ipoprecc	ipoprecu ipoprecu	ipopslsc ipopslsc
ipopslsr ipopslsr	ipopslsu ipopslsu	

MFP Retail

ipcpbopc ipcpbopc	ipcpbopu ipcpbopu	ipcpeopc ipcpeopc
ipcpeopu ipcpeopu	ipcpgmpv ipcpgmpv	ipcprecc ipcprecc
ipcprecu ipcprecu	ipcpslsr ipcpslsr	ipcpslsu ipcpslsu
ipcpbopr ipcpbopr	ipcpeopr ipcpeopr	ipcprecr ipcprecr
ipopbopc ipopbopc	ipopbopu ipopbopu	ipopeopc ipopeopc
ipopeopu ipopeopu	ipopgmpv ipopgmpv	ipoprecc ipoprecc
ipoprecu ipoprecu	ipopslsr ipopslsr	ipopslsu ipopslsu
ipopbopr ipopbopr	ipopeopr ipopeopr	ipoprecr ipoprecr

Example
loadactuals.ksh LoadActualsList.txt 1

Error Information

Table 7–16   Load Actuals Data Error Information

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

Notes

- This script uses the RPAS loadmeasure utility. See the *Oracle Retail Predictive Application Server Administration Guide for the Classic Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Fusion Client* 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 file name.

- The script does not produce an error when records are rejected from the loaded files. These rejected records are logged in the log output of the load process.

Process Actuals Data

Script
processactuals.ksh

Usage
processactuals.ksh

Error Information

Table 7–17   Process Actuals Error Information

Task Name	Error Code	Abort Required?	Description of Error
processactuals	Non-zero value	yes	Errors occurred during the execution of the batch_calc or Batch_Inv rule groups.

Notes

- Script location is:
  - MFP Retail: $MFP_RETAIL_HOME/bin/actualize/processactuals.ksh
  - MFP Cost: $MFP_COST_HOME/bin/actualize/processactuals.ksh
- This script uses the RPAS mace utility. 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* for details on this utility.
- This script runs the batch_calc rule group across all local domains in parallel using the $BSA_MAX_PARALLEL variable (defined in environment.ksh) to cap the number of simultaneous background processes. Upon completion of the batch_calc rule group execution, the script runs the Batch_Inv rule group across all local domains in the same fashion.

Run Batch Forecast

Script
runforecast.ksh

Usage
runforecast.ksh

Error Information

Table 7–18   Run Batch Forecast Error Information

Task Name	Error Code	Abort Required?	Description of Error
runforecast	13	yes	Domain specified by $MFP_MASTERDOMAIN does not exist.
runforecast	Other non-zero value	yes	Errors occurred during the execution of the mace expression to calculate the BUWpStartDt or Batch_FC rule group.

Notes

- Script location is:
  - MFP Retail: $MFP_RETAIL_HOME/bin/forecast/runforecast.ksh
  - MFP Cost: $MFP_COST_HOME/bin/forecast/runforecast.ksh
- This script uses the RPAS mace utility. 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* for details on this utility.
- This script sets the BUWpStartDt measure to the current date, and then executes the Batch_FC rule group across all local domains in parallel using the $BSA_MAX_PARALLEL variable (defined in environment.ksh) to cap the number of simultaneous background processes.

Refresh Workbooks

Script
refresh.ksh

Usage
refresh.ksh

Error Information

Table 7–19   Refresh Workbooks Error Information

Task Name	Error Code	Abort Required?	Description of Error
refresh	13	yes	Domain does not exist.
refresh	Other non-zero value	yes	Running the wbbatch utility produced errors. Further batch processing should not be run until the error has been investigated.

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 the Classic Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Fusion Client* for details on this utility.

- Workbooks should be committed before running 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 running both the auto build and refresh scripts, it is important to run the refresh job first. The refresh updates all MFP 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

Table 7–20   Auto Build Workbooks Error Information

Task Name	Error Code	Abort Required?	Description of Error
autobuild	13	yes	Domain does not exist.
autobuild	Other non-zero value	yes	Running the wbbatch utility produced errors. Further batch processing should not be run until the error has been investigated.

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 the Classic Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Fusion Client* for details on this utility.

■ Auto build entries must be added via the domain before running this script. See the *Oracle Retail Predictive Application Server User Guide for the Classic Client* or the *Oracle Retail Predictive Application Server User Guide for the Fusion 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 MFP workbooks saved in the domain. If the auto build is run first, these workbooks are unnecessarily refreshed.

Load Mapping Measures

Script
loadactuals.ksh

Usage
loadactuals.ksh <measurelist> <maxprocesses>

Table 7–21   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 $MFP_HOME/bin directory. The file name should follow the convention of IP_MFP_transferList.txt.
maxprocesses	Sets the maximum number of export processes to run in parallel.	The default is 1.

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

■ bulylagtx

■ buallagtx

Example
loadactuals.ksh LoadMapList.txt 1

Error Information

Table 7–22   Load Actuals Error Information

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

Notes

- This script uses the RPAS loadmeasure utility. See the *Oracle Retail Predictive Application Server Administration Guide for the Classic Client* or the *Oracle Retail Predictive Application Server Administration Guide for the Fusion Client* 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 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.