

**Oracle® Retail Merchandise Financial Planning**

Operations Guide

Release 13.0.2

December 2008

Copyright © 2008, Oracle. All rights reserved.

Primary Author: Bernadette Goodman

Contributing Author: Tom Virant, Joe Krebs, Scott Dingfelder, Gaurav Marwah

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

## Value-Added Reseller (VAR) Language

### Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

- (i) the software component known as **ACUMATE** developed and licensed by Lucent Technologies Inc. of Murray Hill, New Jersey, to Oracle and imbedded in the Oracle Retail Predictive Application Server - Enterprise Engine, Oracle Retail Category Management, Oracle Retail Item Planning, Oracle Retail Merchandise Financial Planning, Oracle Retail Advanced Inventory Planning and Oracle Retail Demand Forecasting applications.
- (ii) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.
- (iii) the **SeeBeyond** component developed and licensed by Sun Microsystems, Inc. (Sun) of Santa Clara, California, to Oracle and imbedded in the Oracle Retail Integration Bus application.
- (iv) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Store Inventory Management.
- (v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by Business Objects Software Limited ("Business Objects") and imbedded in Oracle Retail Store Inventory Management.
- (vi) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.
- (vii) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.
- (viii) the software component known as **Style Report™** developed and licensed by InetSoft Technology Corp. of Piscataway, New Jersey, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.
- (ix) the software component known as **DataBeacon™** developed and licensed by Cognos Incorporated of Ottawa, Ontario, Canada, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.



---

---

# Contents

|  |     |
|--|-----|
| <b>Preface</b> .....   | ix  |
| Audience.....  | ix  |
| Related Documents .....  | x   |
| Customer Support.....  | x   |
| Review Patch Documentation .....                                   | x   |
| Oracle Retail Documentation on the Oracle Technology Network ..... | x   |
| Conventions .....  | xi  |
| <br>   |     |
| <b>1 Introduction</b>  |     |
| <b>Contents of this Guide</b> .....                                | 1-1 |
| <b>Overview of the Fashion Planning Bundle</b> .....               | 1-1 |
| <b>Architecture of the Template Applications</b> .....             | 1-2 |
| <b>Business Process Flow</b> .....                                 | 1-3 |
| <b>Key Features of MFP</b> .....                                   | 1-3 |
| <b>Skills Needed for Implementation</b> .....                      | 1-4 |
| Applications .....   | 1-4 |
| Technical Concepts .....   | 1-4 |
| <br>   |     |
| <b>2 Implementation Considerations</b>                             |     |
| <b>Historical Data</b> .....                                       | 2-1 |
| <b>Sizing Impacts</b> .....  | 2-1 |
| <b>Partitioning</b> .....  | 2-2 |
| <b>Formatting</b> .....  | 2-2 |
| <b>Plug-ins</b> .....  | 2-2 |
| <b>Patch Considerations</b> .....                                  | 2-3 |
| <b>Batch Scheduling</b> .....                                      | 2-3 |
| <b>Security</b> .....  | 2-3 |
| <b>Internationalization</b> .....                                  | 2-4 |
| <br>   |     |
| <b>3 Build Scripts</b>   |     |
| <b>Installation Dependencies</b> .....                             | 3-1 |
| Environment Setup .....  | 3-1 |
| RPAS Installation .....  | 3-1 |

|  |            |
|--|------------|
| MFP Installation .....   | 3-2        |
| Custom Domain Build.....   | 3-2        |
| Handling Common Hierarchy Files in the Fashion Planning Bundle Applications..... | 3-2        |
| <b>Building the MFP RPAS Domain.....</b>   | <b>3-3</b> |
| Batch Designs.....   | 3-3        |
| Building a Domain.....   | 3-4        |
| Patching a Domain.....   | 3-5        |
| <b>Loading and Extracting Data.....</b>  | <b>3-5</b> |

## 4 Integration

|  |            |
|--|------------|
| <b>Overview of the Fashion Planning Bundle.....</b>                      | <b>4-1</b> |
| Execution Applications .....   | 4-2        |
| Optimization Applications .....  | 4-3        |
| Planning Applications.....   | 4-3        |
| <b>Overview of the MFP Data Flow.....</b>                                | <b>4-4</b> |
| Integration Interface Data Flow Description.....                         | 4-4        |
| From a Merchandising Application to Merchandise Financial Planning.....  | 4-4        |
| From a Data Warehouse Application to Merchandise Financial Planning..... | 4-5        |
| From Item Planning to Merchandise Financial Planning .....               | 4-5        |
| From Merchandise Financial Planning to Assortment Planning.....          | 4-5        |
| From Merchandise Financial Planning to Item Planning .....               | 4-5        |
| <b>Integration Scripts.....</b>  | <b>4-5</b> |
| Batch Designs.....   | 4-6        |
| Load Actuals .....   | 4-6        |
| Export to IP .....   | 4-9        |
| Export to AP.....  | 4-11       |

## 5 Batch Processing

|  |            |
|--|------------|
| <b>Batch Structure Overview.....</b>   | <b>5-1</b> |
| <b>Batch Scheduling.....</b>           | <b>5-1</b> |
| Daily Batch Scripts .....              | 5-2        |
| Weekly Batch Scripts .....             | 5-2        |
| Unscheduled Administration Script..... | 5-3        |
| Batch Environment Scripts .....        | 5-3        |
| <b>Batch Designs.....</b>              | <b>5-4</b> |
| Calendar Hierarchy Load .....          | 5-5        |
| Product Hierarchy Load.....            | 5-6        |
| Location Hierarchy Load .....          | 5-7        |
| Load On Order Data .....               | 5-8        |
| Load Actuals Data.....                 | 5-10       |
| Load IP Data .....                     | 5-13       |
| Process Actuals Data.....              | 5-15       |
| Run Batch Forecast.....                | 5-16       |
| Refresh Workbooks.....                 | 5-17       |
| Auto Build Workbooks.....              | 5-18       |
| Load Mapping Measures .....            | 5-19       |

## List of Figures

|     |   |     |
|-----|---|-----|
| 1-1 | Architecture of the RPAS Template Applications.....               | 1-2 |
| 1-2 | Business Process Workflow .....                                   | 1-3 |
| 4-1 | Conceptual Overview.....  | 4-1 |
| 4-2 | Overview of Fashion Planning Bundle Integration .....             | 4-2 |
| 4-3 | Data Flow for the Integration of MFP with Other Applications..... | 4-4 |

## List of Tables

|      |   |      |
|------|---|------|
| 3-1  | Build Scripts.....                              | 3-3  |
| 3-2  | Building a Domain Error Information .....       | 3-4  |
| 3-3  | Patching a Domain Error Information .....       | 3-5  |
| 4-1  | Integration Script .....                        | 4-6  |
| 4-2  | Load Actuals Error Information .....            | 4-8  |
| 4-3  | Export to IP Error Information.....             | 4-10 |
| 4-4  | Export to AP Error Information.....             | 4-12 |
| 5-1  | Directories Used by Batch Scripts.....          | 5-1  |
| 5-2  | Daily Batch Scripts.....                        | 5-2  |
| 5-3  | Weekly Batch Scripts .....                      | 5-2  |
| 5-4  | Unscheduled Administration Script .....         | 5-3  |
| 5-5  | Calendar Hierarchy Load Error Information.....  | 5-5  |
| 5-6  | Product Hierarchy Load Error Information.....   | 5-6  |
| 5-7  | Location Hierarchy Load Error Information ..... | 5-7  |
| 5-8  | Load Actuals Error Information .....            | 5-8  |
| 5-9  | Load Actuals Data Error Information.....        | 5-12 |
| 5-10 | Load Actuals Data Error Information.....        | 5-14 |
| 5-11 | Process Actuals Error Information .....         | 5-15 |
| 5-12 | Run Batch Forecast Error Information.....       | 5-16 |
| 5-13 | Refresh Workbooks Error Information.....        | 5-17 |
| 5-14 | Auto Build Workbooks Error Information.....     | 5-18 |
| 5-15 | Load Actuals Error Information .....            | 5-19 |

---

---

# Preface

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

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

## Audience

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

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

## Related Documents

For more information, see the following documents in the Oracle Retail Merchandise Financial Planning Release 13.0.2 documentation set:

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

For more information on the Fashion Planning Bundle applications, see the following documents:

- Oracle Retail Assortment Planning documentation
- Oracle Retail Item Planning documentation
- Oracle Retail Item Planning Configured for COE documentation
- Oracle Retail Size Profile Optimization documentation

## Customer Support

<https://metalink.oracle.com>

When contacting Customer Support, please provide:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to recreate
- Exact error message received
- Screen shots of each step you take

## Review Patch Documentation

If you are installing the application for the first time, you install either a base release (for example, 13.0) or a later patch release (for example, 13.0.2). If you are installing a software version other than the base release, be sure to read the documentation for each patch release (since the base release) before you begin installation. Patch documentation can contain critical information related to the base release and code changes that have been made since the base release.

## Oracle Retail Documentation on the Oracle Technology Network

In addition to being packaged with each product release (on the base or patch level), all Oracle Retail documentation is available on the following Web site (with the exception of the Data Model which is only available with the release packaged code):

[http://www.oracle.com/technology/documentation/oracle\\_retail.html](http://www.oracle.com/technology/documentation/oracle_retail.html)

Documentation should be available on this Web site within a month after a product release. Note that documentation is always available with the packaged code on the release date.

# Conventions

The following text conventions are used in this document:

| <b>Convention</b> | <b>Meaning</b>   |
|-------------------|--|
| <b>boldface</b>   | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.         |
| <i>italic</i>     | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.                          |
| monospace         | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |



---

---

# Introduction

Oracle Retail 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: Integration. Overview of the Fashion Planning Bundle and explanation of the MFP data flow and integration scripts.
- Chapter 5: Batch Processing. Explanation of batch scheduling and batch designs.

## Overview of the Fashion Planning Bundle

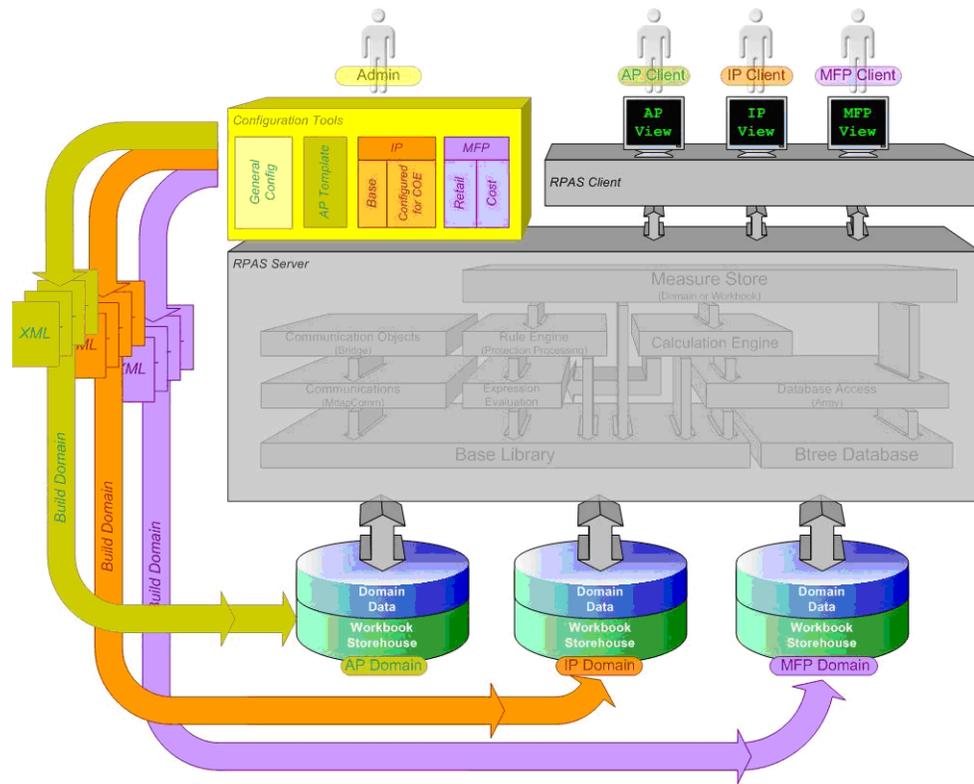
The Fashion Planning Bundle is the integration of Item Planning (IP), Clearance Optimization Engine (COE), Assortment Planning (AP), Merchandise Financial Planning (MFP), and Size Profile Optimization (SPO) as a full-suite planning solution for fashion retailers.

For details on the integration, see [Chapter 4](#).

## Architecture of the Template Applications

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

**Figure 1–1 Architecture of the RPAS Template Applications**



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

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

1. Measures
2. Special expressions
3. Rules
4. Workbook layout

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

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

## Business Process Flow

Figure 1–2 shows a typical workflow for 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
- Oracle Retail Assortment Planning
- Oracle Retail Item Planning
- Oracle Retail Clearance Optimization Engine
- Oracle Retail Merchandise Financial Planning
- Oracle Retail Size Profile Optimization
- Oracle Retail Data Warehouse (RDW)
- Oracle Retail Merchandising System (RMS)

### Technical Concepts

The implementer should understand the following technical concepts:

- UNIX system administration, shell scripts, and job scheduling
- Performance constraints based on the retailer's infrastructure
- Technical architecture for 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

---

---

## Implementation Considerations

The following information needs to be considered before configuring MFP:

- "Historical Data"
- "Sizing Impacts"
- "Partitioning"
- "Formatting"
- "Plug-ins"
- "Patch Considerations"
- "Batch Scheduling"
- "Security"
- "Internationalization"

### Historical Data

It is recommended that 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.

### Sizing Impacts

The following factors can affect size requirements:

- Style-Color—number of style-colors. A style-color is the combination of the style information, such as fleece jacket, and color, such as, navy. 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 can cause 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, men's merchandise could be in a domain, women's merchandise in a domain, and children's merchandise in a domain. When a user is committing data in the men's merchandise domain, this will not affect the users in the women's or children's domains because of the use of partitioning.

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

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

---

---

**Note:** The partitioning level in the 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*.

## Plug-ins

---

---

**Note:** There is no Configuration Tools plugin 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.

## 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 be able to edit any values or commit the workbook. The read-only workbook can be refreshed.

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

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

---

---

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

---

---

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

## Internationalization

Translation is the process of interpreting and adapting text from one language into another. Although the code itself is not translated, components of the application that are translated may include the following, among others:

- Graphical user interface (GUI)
- Error messages

The RPAS platform supports associated solution extensions and solution templates.

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

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

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

---

---

## Build Scripts

This chapter describes the scripts that must be executed to build 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 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\_HOME. The UNIX user performing the installation needs to set up an environmental variable called MFP\_HOME in the user's profile:

```
export MFP_HOME=<full path name to 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

## MFP Installation

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

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. Update the `localdomainlist.cfg` file with the correct paths for the local domains.
5. If needed, update the default environment variables in `environment.ksh`.
6. Execute the `build_mfp.ksh` script: `./build_mfp.ksh`
7. It is expected that the first time `build_mfp.ksh` is executed, an error occurs when it tries to remove the old log file because a log file does not yet exist.

## Handling Common Hierarchy Files 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 rpaInstall, can differentiate between standard and hdr hierarchy files. There is no need for the implementer to make any changes in the domain build process.

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

---



---

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

---



---

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

## Building the MFP RPAS Domain

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

Table 3–1 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 | -2         | yes             | Error when loading users.  |
| build_mfp | -3         | yes             | Error when loading measures.   |
| build_mfp | -4         | yes             | Error in the master domain calculations.                                   |
| build_mfp | -5         | yes             | localdomainlist.cfg file is not found.                                     |
| 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 details on this utility.
- The script also uses the following RPAS utilities: `usermgr`, `mace`, `loadmeasure`, `copydomain`, and `domainprop`. See the *Oracle Retail Predictive Application Server Administration Guide* for details on these utilities.
- All hierarchy and measure files need to be placed in `$MFP_HOME/input` before the execution of this script.
- The script creates four initial users: `ac1`, `bu1`, `mo1`, and `td1`. The passwords are initially set to be the same as the user ID.
- 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_keepformats<br>patch_mfp_deleteformats | -101       | yes             | Error during domain patch. |

### Notes

- The difference between the scripts is whether the formats are updated.
  - The patch\_mfp\_keepformats.ksh script should be run if configuration changes do not affect formatting.
  - The patch\_mfp\_deleteformats.ksh script should be run if the configuration requires formatting updates. This script removes all existing saved formats (template, group, and user).
- The scripts utilize the Configuration Tools rpasInstall utility to build a domain. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.

## Loading and Extracting Data

Data is loaded into MFP using the standard RPAS approach. See the *Oracle Retail Predictive Application Server Administration Guide* for details on formatting the load data files and on the utilities that enable administrators to load data into RPAS. For information on integration scripts, see [Chapter 4](#). For information on any other batch scripts, see [Chapter 5](#).



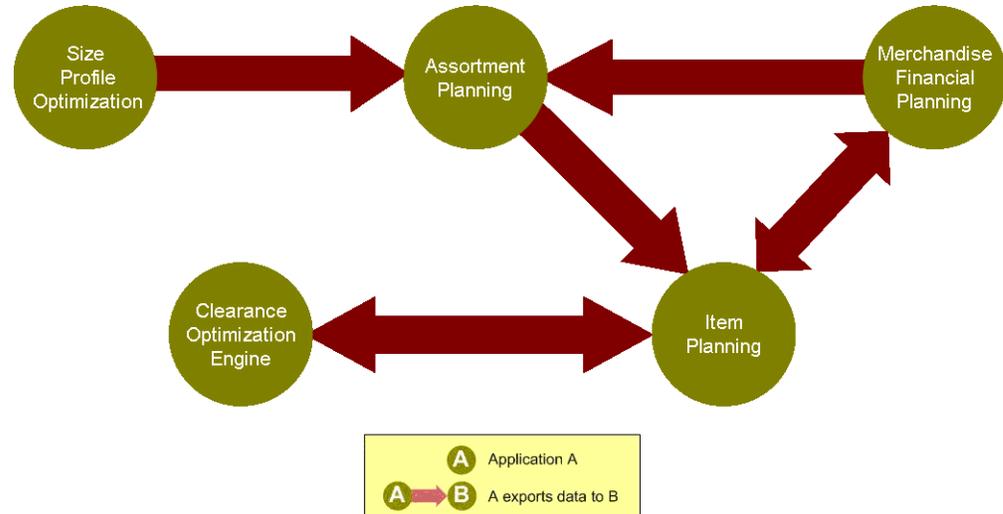
This chapter describes the integration of the applications that are part of the Fashion Planning Bundle.

## 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.

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

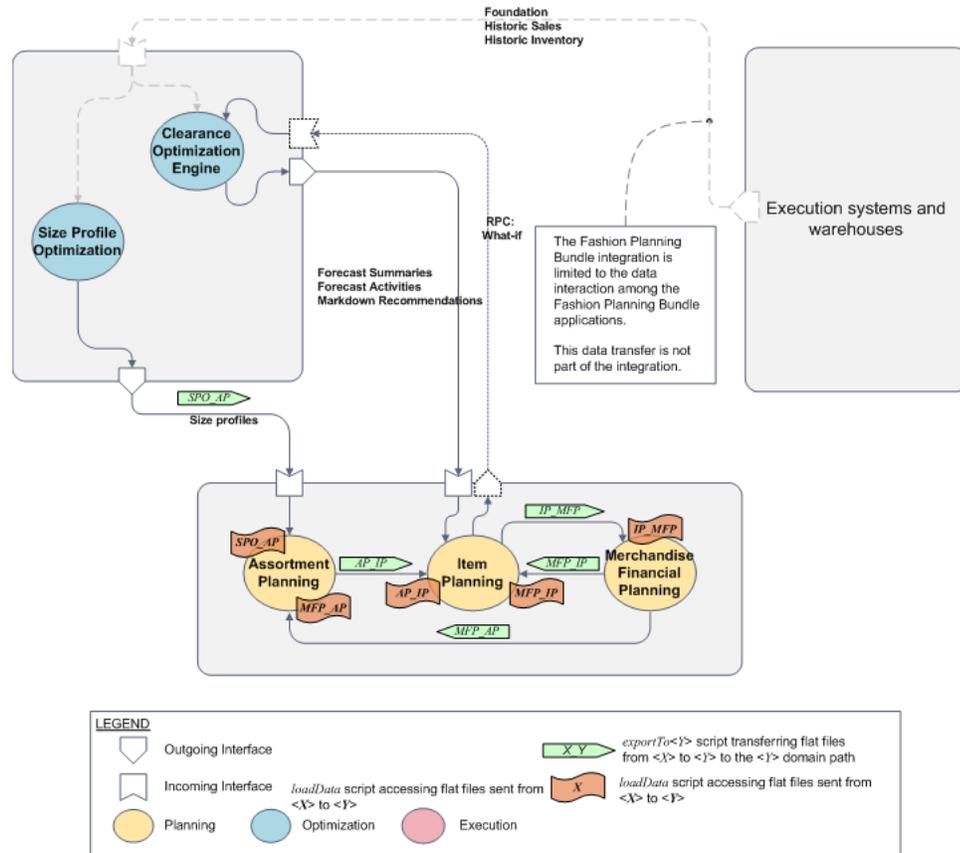
**Figure 4-1** Conceptual Overview



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.

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

---

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

- A data warehouse application provides the initial load of sales and inventory data. For information on what to consider when providing the historical data, see ["Historical Data"](#) in [Chapter 2](#).
- 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 you 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

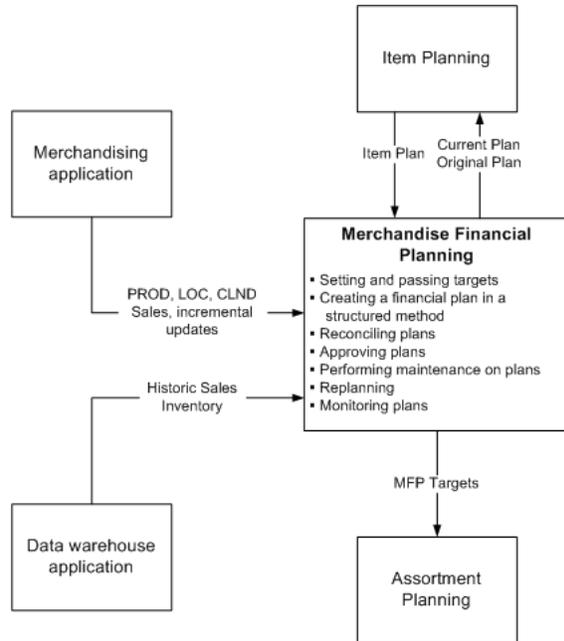
The planning applications take data from the optimization applications and create plans for the retailer.

- AP creates buy plans for the retailer based on size profiles from SPO and MFP targets from MFP. The buy plans feeds into IP.
- IP takes the buy plan from AP and MFP targets 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 to create MFP targets. Based on financial goals, the MFP targets determine how much merchandise should be available in a store in order to meet the goals. MFP targets are fed into AP and IP.

## Overview of the MFP Data Flow

Figure 4–3 shows the integration of MFP with other applications and the flow of data between the applications.

**Figure 4–3 Data Flow for the Integration of MFP with Other Applications**



## Integration Interface Data Flow Description

These descriptions explain the each of the data flows in Figure 4–3.

### From a Merchandising Application to Merchandise Financial Planning

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

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

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.

### From a Data Warehouse Application to Merchandise Financial Planning

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

- Historic sales
- Inventory

### From Item Planning to Merchandise Financial Planning

The following data is exported from IP to MFP:

- Item plan

MFP takes data from IP to create MFP targets.

### From Merchandise Financial Planning to Assortment Planning

The following data is exported from MFP to AP:

- MFP Targets

MFP targets are critical inputs into the AP process. They define the financial goals for that buying period. AP uses these 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.

### 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.

## 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 4–1 lists the integration scripts for MFP. These scripts are located in `$MFP_HOME/bin`.

**Table 4–1 Integration Script**

| Application | Script Name     | Arguments                              |
|-------------|-----------------|--|
| MFP         | loadactuals.ksh | measurelist, maxprocesses              |
| IP          | exportToIP.ksh  | maxprocesses, destination, measurelist |
| AP          | exportToAP.ksh  | maxprocesses, destination, measurelist |

## Batch Designs

This section contains detailed information on the following integration scripts:

- "Load Actuals"
- "Export to IP"
- "Export to AP"

## Load Actuals

### Script

loadactuals.ksh

### Usage

loadactuals.ksh <measurelist> <maxprocesses>

| Argument     | Description   | Notes  |
|--------------|---|--|
| measurelist  | Sets the location of the file which contains the list of measures to be exported. | By default, these files are provided with the package in the <code>\$MFP_HOME/bin</code> directory. The file name should follow the convention of <code>IP_MFP_transferList.txt</code> . |
| 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:

- `ipcpbopc`
- `ipcpbopu`

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

MFP Retail:

- ipcpbopc
- ipcpbopu
- ipcpeopc
- ipcpeopu
- ipcpgmpv
- ipcprecc
- ipcprecu
- ipcpslsr
- ipcpslsu
- ipcpbopr
- ipcpeopr
- ipcprecr
- ipopbopc
- ipopbopu
- ipopeopc
- ipopeopu
- ipopgmpv
- ipoprecc

- ipoprecu
- ipopslsr
- ipopslsu
- ipopbopr
- ipopeopr
- ipoprecr

**Example**

```
loadactuals.ksh IP_MFP_transfer_list.txt 1
```

**Error Information****Table 4–2 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 | 44         | no              | localdomainlist.cfg file is 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 details on this utility.
- The script ignores any missing or empty measure load files.
- All measure files are placed into the domain's input folder.
- If there were no errors during the loading of all measures, the input file is archived into the domain's input/processed directory. A date stamp is appended to the end of the measure file name.
- The script does not produce an error when records are rejected from the loaded files. These rejected records are logged in the log output of the load process.

## Export to IP

### Script

exportToIP.ksh

### Usage

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

| Argument     | Description  | Notes   |
|--------------|--|---|
| maxprocesses | Sets the maximum number of export processes to run in parallel.                                  | The default is 1.   |
| destination  | Sets the path to the input folder of the destination domain where the exported files are placed. | The export script creates the file containing the exported measures in the input folder of the destination domain.  |
| measurelist  | Sets the location of the file which contains the list of measures to be exported.                | By default, these files are provided with the package in the \$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:

- buopbopc buopbopc
- buopeopc buopeopc
- buopgmpv buopgmpv
- buoprecc buoprecc
- buopslsc buopslsc
- buopslsnetr buopslsnetr
- buopslsr buopslsr
- buopbopu buopbopu
- buopeopu buopeopu
- buoprecu buoprecu
- buopslsu buopslsu
- bucpbopu bucpbopu
- bucppeopu bucppeopu
- bucprecu bucprecu
- bucpslsu bucpslsu
- bucpbopc bucpbopc
- bucppeopc bucppeopc
- bucpgmpv bucpgmpv
- bucprecc bucprecc
- bucpslsc bucpslsc

- bucpslsnetr bucpslsnetr
- bucpslsr bucpslsr

MFP Retail:

- bucpbopc bucpbopc
- bucpbopr bucpbopr
- bucpeopc bucpeopc
- bucpeopr bucpeopr
- bucpgmpv bucpgmpv
- bucpmkdr bucpmkdr
- bucprecc bucprecc
- bucprecr bucprecr
- bucpslsr buopbopc
- buopbopc buopbopc
- buopbopr buopbopr
- buopeopc buopeopc
- buopeopr buopeopr
- buopgmpv buopgmpv
- buopmkdr buopmkdr
- buoprecc buoprecc
- buoprecr buoprecr
- buopslsr buopslsr
- bucpbopu bucpbopu
- bucpeopu bucpeopu
- bucprecu bucprecu
- bucpslsu bucpslsu
- buopbopu buopbopu
- buopeopu buopeopu
- buoprecu buoprecu
- buopslsu buopslsu

**Error Information**

**Table 4–3 Export to IP Error Information**

| <b>Task Name</b> | <b>Error Code</b> | <b>Abort Required?</b> | <b>Description of Error</b>          |
|------------------|-------------------|------------------------|--------------------------------------|
| 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 exportdata utility to export measure data from the domain. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.

**Export to AP****Script**

exportToAP.ksh

**Usage**

exportToAP.ksh &lt;maxprocesses&gt; &lt;destination&gt; &lt;measurelist&gt;

| 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. |

**Control File**

MFP\_AP\_transfer\_list.txt: Contains the following list of measures that can be exported as part of this script:

MFP Cost:

- bucpmpv mfcpposgmr
- bucpcecc mfcprecc
- bucpslsr mfcpslsr
- bucpcecu mfcprecu
- bucpslsu mfcpslsu

MFP Retail:

- bucpmpv mfcpposgmr
- bucpcecc mfcprecc
- bucpslsr mfcpslsr
- bucpcecu mfcprecu
- bucpslsu mfcpslsu
- bucpcecr mfcprecr

**Error Information****Table 4–4 Export to AP Error Information**

| <b>Task Name</b> | <b>Error Code</b> | <b>Abort Required?</b> | <b>Description of Error</b>          |
|------------------|-------------------|------------------------|--------------------------------------|
| 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 exportdata utility to export measure data from the domain. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.

---



---

## Batch Processing

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

### Batch Structure Overview

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

**Table 5–1** *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<br>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/batch` 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 5–2 lists information on the daily batch script. For more information on the Load Actuals and Export to IP scripts, see "Integration Scripts" in Chapter 4.

**Table 5–2 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 5–3 lists information on the weekly batch scripts.

**Table 5–3 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<br><b>Note:</b> This script is optional.           | refresh.ksh        | workbook        | Generate sales forecast   |
| Auto build workbooks placed on queue<br><b>Note:</b> 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 5-4 lists information on the unscheduled administration script.

**Table 5-4** *Unscheduled Administration Script*

| Description  | Script Name     | Batch Directory | Dependency |
|--------------|-----------------|-----------------|------------|
| Load mapping | loadactuals.ksh | bin             | None       |

## Batch Environment Scripts

These scripts are included in the other batch scripts to control logging and set environment variables.

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

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

- `export 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_EXPORT_TOAP=$MFP_HOME/exportToAP`
- `export MFP_INPUTHOME=$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:

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

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

## Calendar Hierarchy Load

### Script

loadhier.ksh

### Usage

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

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

### Example

loadhier.ksh clnd 10000 10

### Error Information

**Table 5–5 Calendar Hierarchy Load Error Information**

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

### Notes

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

## Product Hierarchy Load

### Script

loadhier.ksh

### Usage

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

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

### Example

loadhier.ksh prod 10000 10

### Error Information

**Table 5–6 Product Hierarchy Load Error Information**

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

### Notes

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

## Location Hierarchy Load

**Script**  
loadhier.ksh

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

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

**Example**  
loadhier.ksh loc 10000 10

### Error Information

**Table 5–7 Location Hierarchy Load Error Information**

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

### Notes

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

## Load On Order Data

### Script

loadactuals.ksh

### Usage

loadactuals.ksh <measurelist> <maxprocesses>

| Argument     | Description   | Notes   |
|--------------|---|---|
| measurelist  | Sets the location of the file which contains the list of measures to be exported. | By default, these files are provided with the package in the \$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 5–8 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 details on this utility.
- The script ignores any missing or empty measure load files.
- All measure files are placed into the domain's input folder.
- If there were no errors during the loading of all measures, the input file is archived into the domain's input/processed directory. A date stamp is appended to the end of the measure file name.
- The script does not produce an error when records are rejected from the loaded files. These rejected records are logged in the log output of the load process.

## Load Actuals Data

### Script

loadactuals.ksh

### Usage

loadactuals.ksh <measurelist> <maxprocesses>

| Argument     | Description   | Notes   |
|--------------|---|---|
| measurelist  | Sets the location of the file which contains the list of measures to be exported. | By default, these files are provided with the package in the \$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
- drtyslclrr
- drtyslclru
- drtyslclrc
- drtyslpror
- drtyslprou
- drtyslproc
- drtyslregr
- drtyslregu
- drtyslregc
- drtydevalc
- buwpmatp
- drtyshrinkc

- drtyshrinku
- drtymiscinc
- drtymiscinu
- drtymiscoutc
- drtymiscoutu

## MFP Retail:

- drtybopclrc
- drtybopclru
- drtybopclrr
- drtybopregc
- drtybopregu
- drtybopregr
- drtyeopclrc
- drtyeopclru
- drtyeopclrr
- drtyeopregc
- drtyeopregu
- drtyeopregr
- drtyrecc
- drtyrecu
- drtyrecr
- drtyslclrr
- drtyslclru
- drtyslpror
- drtyslprou
- drtyslsregr
- drtyslsregu
- buwpvatp
- drtyshrinkr
- drtyshrinku
- drtymiscinr
- drtymiscinu
- drtymiscoutr
- drtymiscoutu
- drtymkdr
- drtymkupr

**Example**

loadactuals.ksh LoadOnOrderList.txt 1

**Error Information****Table 5–9 Load Actuals Data Error Information**

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

**Notes**

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

## Load IP Data

### Script

loadactuals.ksh

### Usage

loadactuals.ksh <measurelist> <maxprocesses>

| Argument     | Description   | Notes   |
|--------------|---|---|
| measurelist  | Sets the location of the file which contains the list of measures to be exported. | By default, these files are provided with the package in the \$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 5–10 Load Actuals Data Error Information**

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

**Notes**

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

**Process Actuals Data****Script**

processactuals.ksh

**Usage**

processactuals.ksh

**Error Information****Table 5–11 Process Actuals Error Information**

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

**Notes**

- This script uses the RPAS mace utility. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The script can run the batch\_calc group.

## Run Batch Forecast

**Script**

runforecast.ksh

**Usage**

runforecast.ksh

**Error Information****Table 5–12 Run Batch Forecast Error Information**

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

**Notes**

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

## Refresh Workbooks

### Script

```
refresh.ksh
```

### Usage

```
refresh.ksh <maxprocesses>
```

### where

processes is the maximum number of workbook refresh processes to run in parallel.

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

### Example

```
refresh.ksh 10
```

### Error Information

**Table 5–13 Refresh Workbooks Error Information**

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

### Notes

- This is an optional script. It enables all workbooks to be kept current with the elapsed settings and prevents stale data from being committed.
- The script uses the RPAS wbatch utility to refresh workbooks in the auto build queue. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- Workbooks should be committed before 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 <maxprocesses>

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

### Example

autobuild.ksh 10

### Error Information

**Table 5–14 Auto Build Workbooks Error Information**

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

### Notes

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

| 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 5–15 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 details on this utility.
- The script ignores any missing or empty measure load files.
- All measure files are placed into the domain's input folder.
- If there were no errors during the loading of all measures, the input file is archived into the domain's input/processed directory. A date stamp is appended to the end of the measure file name.
- The script does not produce an error when records are rejected from the loaded files. These rejected records are logged in the log output of the load process.

