

# **Oracle® Retail Demand Forecasting**

Implementation Guide

Release 13.3.2

March 2013

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Primary Author: Melissa Artley

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

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

## Value-Added Reseller (VAR) Language

### Oracle Retail VAR Applications

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

(i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

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

(iii) the software component known as **Access Via**<sup>™</sup> licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(iv) the software component known as **Adobe Flex**<sup>™</sup> licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

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

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



---

---

# Contents

<b>Send Us Your Comments</b> .....	xiii
<b>Preface</b> .....	xv
Audience .....	xv
Documentation Accessibility .....	xv
Related Documents .....	xv
Supplemental Documentation .....	xvi
Customer Support .....	xvi
Review Patch Documentation .....	xvi
Oracle Retail Documentation on the Oracle Technology Network .....	xvi
Conventions .....	xvii
<b>1 Introduction</b>	
<b>Contents of this Guide</b> .....	1-1
<b>RDF and the Oracle Retail Enterprise</b> .....	1-2
<b>RDF Business Process Workflow</b> .....	1-3
Parameter Setup .....	1-4
Data Pre-processing .....	1-4
Forecast Generation .....	1-5
Exception Processing and Automatic Approval .....	1-5
User Review and Approval .....	1-5
Ongoing Forecast Assessment .....	1-5
<b>Key Features of RDF</b> .....	1-5
<b>Skills Needed for Implementation</b> .....	1-6
Technical Consultant Role .....	1-6
Application / Business Consultant Role .....	1-6
<b>2 Implementation Considerations</b>	
<b>Input Data</b> .....	2-1
<b>Hardware Space Impacts</b> .....	2-2
<b>Domain Partitioning</b> .....	2-2
<b>Patch Considerations</b> .....	2-3
Patching Process .....	2-3
<b>Batch Scheduling</b> .....	2-3
<b>Security</b> .....	2-3

Internationalization .....	2-4
<b>3 Integration</b>	
<b>Integrated Inventory Planning Suite Data Flow</b> .....	3-2
<b>RDF Supporting RMS Replenishment and Allocation</b> .....	3-2
From RMS to RDF .....	3-3
From RDF to RMS .....	3-3
<b>RDF Data Flow with PPO and RPO</b> .....	3-3
<b>APC-RPO, RPO, RDF Integration</b> .....	3-3
From APC-RPO to RPO.....	3-4
From APC-RPO to RDF.....	3-4
From RPO to RDF .....	3-4
From RDF to RPO .....	3-4
<b>RDF to APC-RO Integration</b> .....	3-4
Initial Load Process.....	3-5
Usage.....	3-6
<b>Integration Script</b> .....	3-7
<b>4 ODI Integration</b>	
<b>ODI Integration</b> .....	4-1
Measure Data Integration .....	4-1
APC-RPO to RPO Package .....	4-1
Data Mapping for APC-RPO to RPO Package .....	4-1
APC-RPO to RDF Package.....	4-2
Data Mapping for APC-RPO to RDF Package.....	4-2
RDF to RPO Package .....	4-2
Data Mapping for RDF to RPO Package .....	4-2
RPO to RDF Package .....	4-2
Data Mapping for RPO to RDF Package .....	4-2
<b>5 Installation Consideration</b>	
<b>Installation Dependencies</b> .....	5-1
RPAS Installation .....	5-1
RPAS Client Installation.....	5-1
RDF Installation.....	5-2
RDF Taskflow for the RPAS Fusion Client.....	5-2
<b>Environment Variables</b> .....	5-2
<b>Files Needed to Build the RDF Domain</b> .....	5-2
Standard RPAS Hierarchy Files .....	5-3
Calendar (CLND) Hierarchy File .....	5-3
Product (PROD) Hierarchy File .....	5-4
Location (LOC) Hierarchy File.....	5-5
Required Data Files.....	5-5
Optional Data Files .....	5-6
Output from RDF to RMS and Retail Analytics .....	5-6

## 6 Configuration Considerations

<b>Forecasting Calendar Hierarchy Requirement</b> .....	6-2
<b>Forecasting Limitations Using the Partition Hierarchy</b> .....	6-2
Causal Forecasting at Source Levels .....	6-2
Loc Hierarchy Limitation.....	6-2
Forecasting Pre-Configuration Data Requirements .....	6-2
Source Data .....	6-2
Plan Data .....	6-2
Spreading Profiles and Seasonal Profiles .....	6-2
<b>Registering the RdfFunctions Library</b> .....	6-3
<b>Editing Forecast Level Parameters</b> .....	6-3
Level Labels.....	6-4
Forecast Methods .....	6-5
About Causal Forecasting.....	6-5
<b>Autogenerating Hierarchies, Measures, Rules and Workbook Templates</b> .....	6-6
<b>Deleting a Forecast Level</b> .....	6-6
<b>Configuring the Cloning Administration Workbook</b> .....	6-6
<b>Editing the RDF GA Configuration</b> .....	6-6
<b>RDF Non-modifiable Hierarchies</b> .....	6-7
Calendar (CLND) .....	6-7
Product (PROD).....	6-7
Location (LOC) .....	6-7

## 7 Batch Processing

<b>About RDF Batch Scripts</b> .....	7-2
RDF Binaries .....	7-2
<b>PreGenerateForecast</b> .....	7-3
PreGenerateForecast Usage .....	7-4
<b>generate</b> .....	7-5
Usage.....	7-5
Return Codes .....	7-5
<b>RDFvalidate</b> .....	7-6
Usage .....	7-6
RDF Validation.....	7-7
Executable Only .....	7-9
Promote Validation.....	7-10
<b>UpdateFnhbiRdf</b> .....	7-11
Usage.....	7-11

## 8 AutoSource

<b>Inputs to AutoSource Binary</b> .....	8-2
<b>AutoSource Measures</b> .....	8-3
Optimal Source Levels.....	8-3
Pick Optimal Level.....	8-3
Usage.....	8-4

## 9 Forecast Approval Alerts

Alerts.....	9-1
Step 0.....	9-1
Step 1 (Option 1).....	9-1
Step 1 (Option 2).....	9-1
Step 2.....	9-2
Step 3.....	9-2

## 10 Adding New Local Domains

loadCurveParameters.ksh.....	10-1
loadRDFParameters.ksh.....	10-1

## 11 Internationalization

Translation.....	11-1
------------------	------

## A RPAS and RDF Integration with RMS

Environment Variable Setup.....	A-2
RDF Transformation Programs.....	A-3
Common Program for All Transformations.....	A-3
Usage.....	A-3
Transformations of Merchandise Hierarchy Data.....	A-4
Transformations of Location Hierarchy Data.....	A-5
Transformations of Calendar Hierarchy Data.....	A-5
Transformations of Daily Sales and Issues Data.....	A-6
Transformations of Weekly Sales and Issues Data.....	A-6
Transformations of Store Open Date Data.....	A-6
Transformations of Store Close Date Data.....	A-6
Transformations of Out of Stock Indicator Data.....	A-7
RDF Transformation Matrix.....	A-8
Loading Transformed RMS Data into RDF.....	A-10
Common Programs for Extracts.....	A-11
Extract of Forecast Data for RMS.....	A-11
Editing for Simple Domains.....	A-12
Load of Extracted Forecast Data and Standard Deviations to RMS.....	A-12
Extract of Diff Profile Data for Allocation.....	A-13
Extract of Store Grade Data for RMS.....	A-14
RDF Extract Matrix.....	A-15
Internationalization Considerations for RETL.....	A-15
Calendar Data.....	A-15
The Unassigned Value in Schemas.....	A-16

---

---

# List of Figures

1-1	RDF and the Oracle Retail Enterprise .....	1-3
1-2	Business Process Workflow .....	1-4
3-1	Integrated Inventory Planning Suite Data Flow .....	3-2
3-2	RDF Data Flow with PPO and RPO .....	3-3
3-3	APC-RPO, RPO, RDF Integration.....	3-3
7-1	Overview of the Batch Forecast Generation Process .....	7-1



---

---

## List of Tables

2-1	Required Data Files.....	2-1
3-1	Integration Script .....	3-7
4-1	APC-RPO to RPO Data .....	4-2
4-2	APC-RPO to RDF Data.....	4-2
4-3	RDF to RPO Data .....	4-2
4-4	RPO to RDF Data .....	4-2
5-1	Calendar Hierarchy Fields.....	5-3
5-2	Product Hierarchy Fields.....	5-4
5-3	Location Hierarchy Fields.....	5-5
5-4	Examples of Required Data Files.....	5-6
6-1	Forecast Level Parameters .....	6-3
6-2	Non-Modified Hierarchies .....	6-7
7-1	Internal Validation Performed by the Plug-in and RDFvalidate utility .....	7-7
7-2	Executable Only .....	7-9
8-1	AutoSource Binary Input Descriptions.....	8-2



---

---

## Send Us Your Comments

Oracle Retail Demand Forecasting Implementation Guide, Release 13.3.2.

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

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

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

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

---

---

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

---

---

Send your comments to us using the electronic mail address: [retail-doc\\_us@oracle.com](mailto:retail-doc_us@oracle.com)

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

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

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



---

---

# Preface

The Oracle Retail Demand Forecasting (RDF) Implementation Guide describes post-installation tasks that need to be performed in order to bring RDF online and ready for production use.

## Audience

This Implementation Guide is intended for the RDF application integrators and implementation staff, as well as the retailer's IT personnel. This guide is also intended for business analysts who are looking for information about processes and interfaces to validate the support for business scenarios within RDF and other systems across the enterprise.

## 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 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 Demand Forecasting Release 13.3.2 documentation set:

- *Oracle Retail Demand Forecasting Release Notes*
- *Oracle Retail Demand Forecasting Installation Guide*
- Oracle Retail Predictive Application Server documentation

## Supplemental Documentation

The following document is available through My Oracle Support at the following URL:

<https://support.oracle.com>

### **Oracle Retail Demand Forecasting 13.3.2 Cumulative Fixed Issues (Note ID 1527992.1)**

This document details the fixed issues and defects for all RDF, Curve, and Grade patch releases prior to and including the current release.

## 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 recreate
- Exact error message received
- Screen shots of each step you take

## Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 13.2) or a later patch release (for example, 13.2.3). If you are installing the base release, additional patch, and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

## Oracle Retail Documentation on the Oracle Technology Network

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

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

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

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

# Conventions

The following text conventions are used in this document:

<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 Demand Forecasting (RDF) is a statistical and promotional forecasting solution. It uses state-of-the-art modeling techniques to produce high quality forecasts with minimal human intervention. Forecasts produced by the RDF system enhance the retailer's supply-chain planning, allocation, and replenishment processes, enabling a profitable and customer-oriented approach to predicting and meeting product demand.

Today's progressive retail organizations know that store-level demand drives the supply chain. The ability to forecast consumer demand productively and accurately is vital to a retailer's success. The business requirements for consumer responsiveness mandate a forecasting system that more accurately forecasts at the point of sale, handles difficult demand patterns, forecasts promotions and other causal events, processes large numbers of forecasts, and minimizes the cost of human and computer resources.

Forecasting drives the business tasks of planning, replenishment, purchasing, and allocation. As forecasts become more accurate, businesses run more efficiently by buying the right inventory at the right time. This ultimately lowers inventory levels, improves safety stock requirements, improves customer service, and increases the company's profitability.

The competitive nature of business requires that retailers find ways to cut costs and improve profit margins. The accurate forecasting methodologies provided with RDF can provide tremendous benefits to businesses.

For a more detailed overview of the functionality within RDF, see the *Oracle Retail Demand Forecasting User Guide*.

## Contents of this Guide

This implementation guide addresses the following topics:

- [Chapter 1, "Introduction"](#): Overview of the RDF 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, "Integration"](#): Overview of integration and explanation of the RDF data flow and integration script.
- [Chapter 4, "ODI Integration"](#): Overview of the integration of APC-RPO, RPO, and RDF using Oracle Data Integrator (ODI).
- [Chapter 5, "Installation Consideration"](#): Information for the setup that must be done prior to building the RDF domain and installing RDF.

- [Chapter 6, "Configuration Considerations"](#): Information on the functional changes or enhancements that can be made for RDF.
- [Chapter 7, "Batch Processing"](#): Information on the RDF batch forecast process.
- [Chapter 8, "AutoSource"](#): Information on the AutoSource utility.
- [Chapter 9, "Forecast Approval Alerts"](#): Information on the usage and configuration of Forecast Approval Alerts.
- [Chapter 10, "Adding New Local Domains"](#): Information on the process of adding new local domains.
- [Chapter 11, "Internationalization"](#): Translations provided for RDF.
- [Appendix A, "RPAS and RDF Integration with RMS"](#): RMS to RDF transformation programs, RDF to RMS extract programs, Grade (RPAS) to RMS extract programs, and Curve (RPAS) to Allocation extract programs.

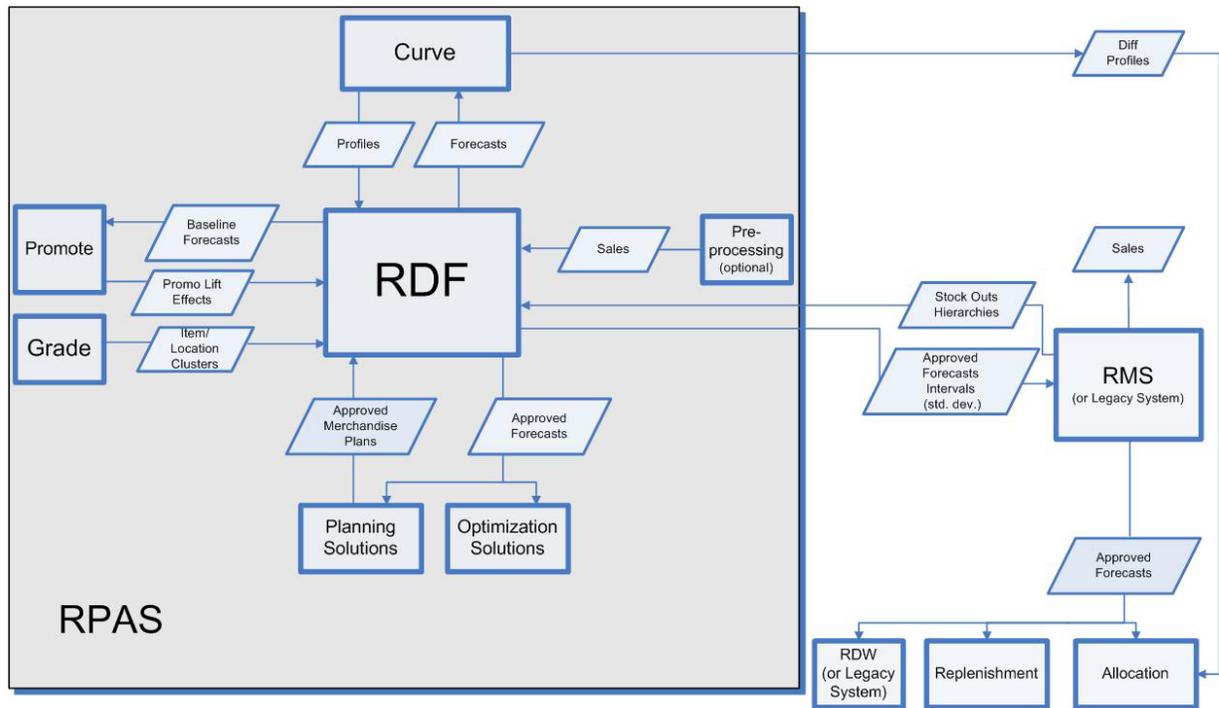
## RDF and the Oracle Retail Enterprise

Oracle Retail has designed a forecasting solution separate from replenishment, allocation or planning. In order to provide a single version of the truth, it is crucial to free up the user's time and supply the tools to focus on the analysis of forecast exceptions, historical data, and different modeling techniques. This empowers the user to make better decisions, thus improving overall accuracy and confidence in the results of the forecast demand.

Within the Oracle Retail Enterprise, Oracle Retail Merchandising System (RMS) supplies RDF with Point-of-Sale (POS) and hierarchy data that is used to create a forecast. Once the forecast is approved, it is exported to RMS in order to calculate a recommended order quantity. Forecasts can also be utilized (no export process required) in any Retail Predictive Application Server (RPAS) solution to support merchandise, financial, collaborative, and price planning processes.

[Figure 1-1](#) shows the interaction between RDF, RPAS, and other applications.

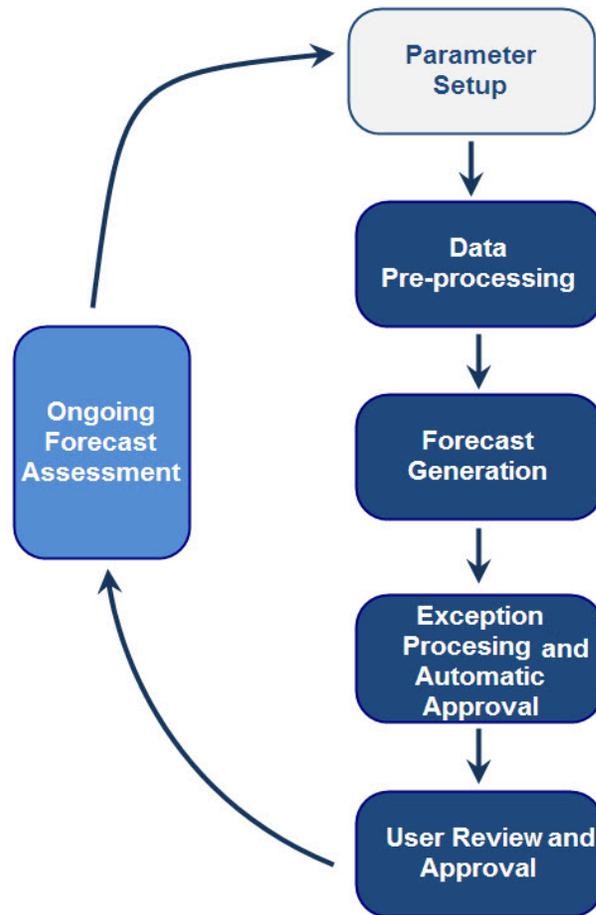
Figure 1-1 RDF and the Oracle Retail Enterprise



## RDF Business Process Workflow

One of the challenges in retail forecasting is the data volumes. The RDF business process focusses on automation, accuracy and lends itself to easy analysis. RDF focuses on automation by automatically selecting best forecast methods and parameters as well as by automatic approval of forecasts that don't meet any exception criteria. Also, it allows users to analyze and manually approve forecasts. Forecast scorecarding allows users to monitor forecast accuracy over time and re-tune settings if necessary.

Figure 1-2 illustrates the RDF business process workflow.

**Figure 1–2 Business Process Workflow**

## Parameter Setup

Following the initial setup, these parameters are not set on a scheduled basis, but are updated as needed.

- Preprocessing and alert parameter setup
- Sets forecast methods, parameters, and specifies source levels
- Sets history start and end dates

## Data Pre-processing

For data pre-processing, RDF:

- Corrects for lost sales due to stock-outs
- Cleanses data for effects of promotions and short-term price changes (optional)
- Manual data-scrubbing (fake history and user history overrides)

## Forecast Generation

For forecast generation, RDF:

- Computes demand parameters (seasonality, level, trend)
- Optimizes exponential smoothing parameters
- Allows you to select best forecast method for SKU/location or use Automatic Exponential Smoothing (AutoES)

## Exception Processing and Automatic Approval

For exception processing and automatic approval, RDF:

- Evaluates forecast for exceptions based on specific alert criteria
- Automatically approves non-alerted forecasts
- Allows you to review flagged exception forecasts

## User Review and Approval

For user review and approval, RDF:

- Reviews and analyzes forecasts, allowing for overrides if necessary
- Approves forecasts

## Ongoing Forecast Assessment

For ongoing forecast assessment, RDF:

- Assesses user overrides versus system forecasts against actuals
- Assesses forecast quality and user adoption
- Retunes parameter settings as needed

## Key Features of RDF

RDF provides the following features:

- Pre-processing to correct for stock-outs and other data anomalies
- Generation of forecasts
- Optimizes forecasting methods and exponential smoothing parameters
- Selects best forecasting methods and parameters to overcome data sparsity and reliability issues
- Generation of alerts and automatic approval of forecasts
- Allows you to facilitate review of analysis and approval of forecasting

## Skills Needed for Implementation

A typical RDF implementation team has technical and application/business consultants in addition to other team members.

The technical and application/business consultants need to have a high level understanding of other applications that RDF can integrate with, which include:

- Advanced Inventory Planning (AIP)
- Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
- Analytic Parameter Calculator for Replenishment Optimization (APC-RO)
- Promotion Planning and Optimization (PPO)
- Replenishment Optimization (RO)
- Retail Merchandising System (RMS)
- Retail Price Optimization (RPO)

In addition, both technical and application/business consultants need to have an understanding of RPAS, its calculation engine, and multi-dimensional database concepts.

---

---

**Note:** Staffing models and roles and responsibilities may vary from project to project, but following is a recommendation based on best practices.

---

---

### Technical Consultant Role

The technical consultant is usually responsible for the following key areas in addition to other activities:

- Interface work
- Batch scripting
- RPAS/RDF domain partitioning

---

---

**Note:** The technical consultant should also be well versed in Unix, Shell scripting, and batch schedulers.

---

---

### Application / Business Consultant Role

The application/business consultant is responsible for:

- Designing and configuring alerts
- Configuring pre-processing rules
- Any workflow/workbook customizations needed to meet retailers business process needs

---

---

**Note:** The application consultant should have a strong understanding of RPAS configuration rule language, RPAS configuration tools, RDF plug-in, and have experience configuring solutions on RPAS.

---

---

---



---

## Implementation Considerations

The following information needs to be considered before implementing RDF:

- [Input Data](#)
- [Hardware Space Impacts](#)
- [Domain Partitioning](#)
- [Patch Considerations](#)
- [Batch Scheduling](#)
- [Security](#)
- [Internationalization](#)

### Input Data

RDF uses the following required data:

**Table 2–1 Required Data Files**

File	Filename	Intersection
Weekly Regular Sales	rsal.ovr	item/store/week
Weekly Promotional sales	psal.ovr	item/store/week
Weekly Clearance sales	csal.ovr	item/store/week
Daily sales (if forecasting at day level)	dpos.ovr	item/store/day
Out of stock indicator	outind.ovr	item/store/week

The following are optional files:

- Promotion history
- Format and file details as specified during implementation time
- Filename varies by promotion name and intersection

## Hardware Space Impacts

The following factors can affect hardware space requirements:

- SKU—Number of distinct items.
- Store—Number of physical, Web, and other distinct retail outlets.
- 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.

## Domain Partitioning

Partitioning is done to avoid competition 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 RDF 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:** Domain partitioning is supported only along Product hierarchy (PROD). This is a standard RPAS hierarchy. Also source levels have to be below partition dimension, that is, if using Dept for source level forecasting, you have to partition at or below Dept.

---

---

In the GA configuration, group is a dimension label. The group dimension is a regular dimension in the product hierarchy, which the customer can rename or delete.

One of the major purposes of partitioning in RDF is to facilitate the parallelization of the batch process.

The wise selection of partition intersections can significantly reduce the batch time. Partition intersection selection should also consider business needs in such a way that contention issues are minimized.

## Patch Considerations

With a new release, there are two types of patches that can affect the RDF 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 straight forward 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.

## Patching Process

Before patching an RDF domain, confirm that the necessary RPAS client, server, and Configuration Tools patch updates have been successfully applied. Refer to the RPAS Installation Guide for RPAS installation instructions.

## Batch Scheduling

RDF batch is typically scheduled to run end of day/end of week with the most updated feeds of sales history and foundation data. Some tasks or batch processes can be run on adhoc or as needed basis.

Following is a list of typical RDF batch tasks and scheduling considerations:

- Daily or weekly activities:
  - Hierarchy Load
  - Data Load
  - Pre-processing
  - Forecast Generation
  - Alert Generation
- Commit batch (committing workbooks saved to be committed later)
- Auto Workbook build
- Adhoc/as needed:
  - AutoSource

## 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 cannot edit any values or commit the workbook. The read-only workbook can be refreshed.

When users save workbooks, they assign one of three access permissions:

- 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*.

## Internationalization

For more information on translation for RDF, see [Chapter 11, "Internationalization"](#).

---

---

## Integration

This chapter describes the interaction between RDF and other applications and the script used to load demand data.

RDF is integrated with the following Oracle Retail applications listed by group:

- [Integrated Inventory Planning Suite Data Flow](#)
  - Advanced Inventory Planning (AIP)
  - Analytic Parameter Calculator for Replenishment Optimization (APC-RO)
  - Replenishment Optimization (RO)
- [RDF Supporting RMS Replenishment and Allocation](#)
  - Retail Merchandising System (RMS)
- [RDF Data Flow with PPO and RPO](#)
  - Promotion Planning and Optimization (PPO)
  - Retail Price Optimization (RPO)
- [APC-RPO, RPO, RDF Integration](#)
  - Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
  - Retail Price Optimization (RPO)
- [RDF to APC-RO Integration](#)
  - Analytic Parameter Calculator for Replenishment Optimization (APC-RO)
- [Integration Script](#)

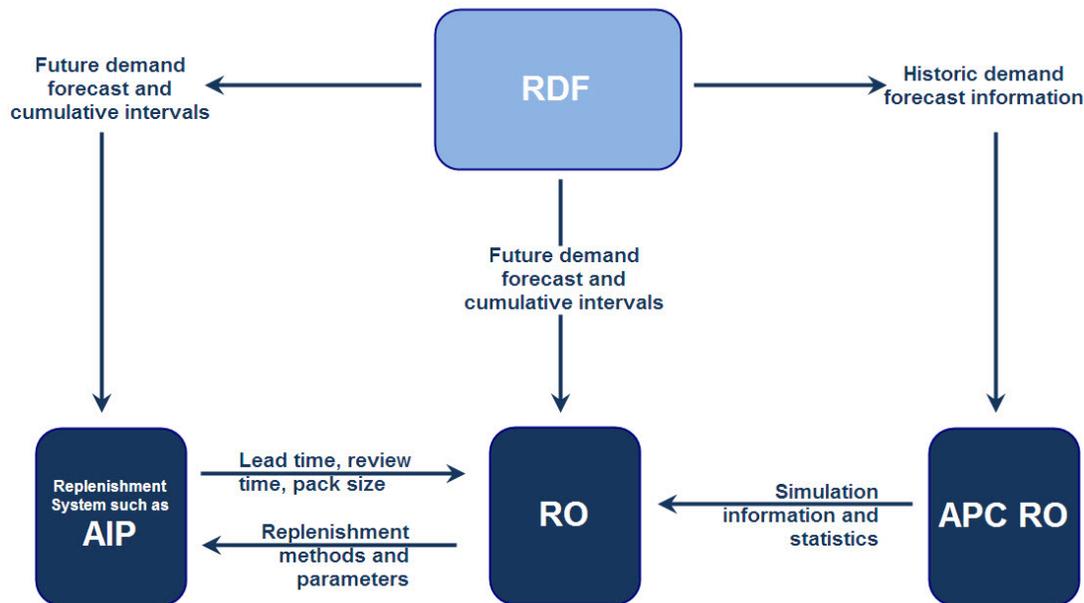
Integration scripts are used for moving data between applications.

## Integrated Inventory Planning Suite Data Flow

Figure 3–1 shows the integration of the Integrated Inventory Planning Suite applications and the flow of data among those applications. Note that Figure 3–1 shows a replenishment system. This can be AIP or any other replenishment system. The demand forecasting application can be RDF or any other forecasting system. RDF forecasts are used as input to RO for simulation-determined replenishment parameters. RDF forecasts and associated statistics are used by AIP to plan time-phased replenishment.

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 3–1 Integrated Inventory Planning Suite Data Flow



## RDF Supporting RMS Replenishment and Allocation

**Note:** For detailed information about the RMS and RDF interface, see the following:

- [Appendix A, "RPAS and RDF Integration with RMS"](#)
- *Oracle Retail Merchandising System Operations Guide, Volume 1*

RDF integrates with Retail Merchandising System (RMS) to receive foundation data. In addition, it also sends weekly and daily forecasts to RMS (replenishment and allocation). These descriptions explain the data flows between RMS and RDF:

## From RMS to RDF

- Product hierarchy
- Location hierarchy
- Calendar hierarchy

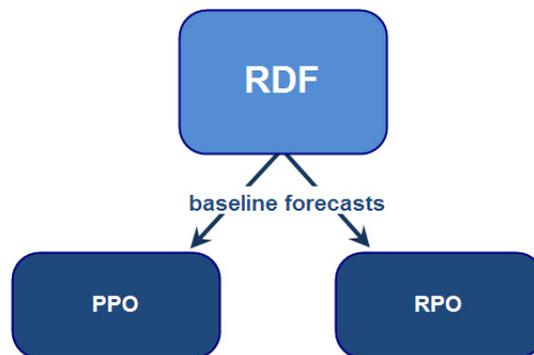
## From RDF to RMS

- Weekly and daily forecasts and cumulative intervals

## RDF Data Flow with PPO and RPO

RDF sends baseline forecasts to PPO and RPO.

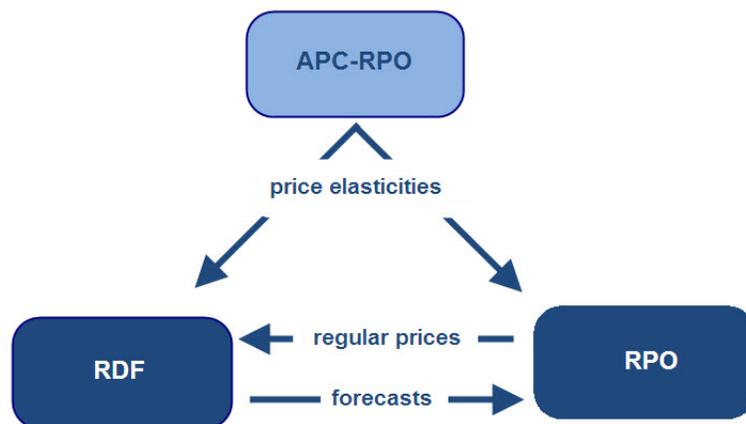
*Figure 3–2 RDF Data Flow with PPO and RPO*



## APC-RPO, RPO, RDF Integration

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

*Figure 3–3 APC-RPO, RPO, RDF Integration*



## From APC-RPO to RPO

- Sends item and cross item elasticities of items. RPO uses these elasticities to optimize prices.
- Sends maximum and minimum historical prices of items. RPO uses this data to optimize prices.
- Sends anchor prices of items. Anchor prices are the baseline prices that APC-RPO uses to calculate price elasticity. RPO uses the anchor prices to calculate price drift metrics.
- Sends maximum price increase and decrease percentages, self-item elasticity standard errors, and self-item elasticities t-statistics. RPO uses the maximum price increase and decrease percentages to setup up the default minimum and maximum price percentage change. Meanwhile, the RPO user can refer to the self-item elasticity standard error and t-statistics to adjust the price constraint.

## From APC-RPO to RDF

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

## From RPO to RDF

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

## From RDF to RPO

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

## RDF to APC-RO Integration

This section describes the integration process between RDF and APC-RO.

RDF's forecasts are important inputs for APC-RO. Forecasts are provided in two ways:

- Initial load
- Weekly updates

At initial load, a rolling 52 forecasts are generated and exported to APC-RO, each of the forecasts starts one week after another.

Secondly, as weekly updates, RDF exports currently forecast to APC-RO. The main purpose of the scripts is to generate RDF forecast in RDF GA domain and export the forecasts from a RDF domain and convert the forecasts into the format required by APC-RO.

Additionally, APC-RO provides to RDF a list of item/stores, and RDF only exports the forecasts for those item/stores to APC-RO.

## Initial Load Process

The initial load process of RDF to APC-RO is as follows:

1. Loading the forecast export mask at item/store from APC-RO into a Boolean measure named `apcroexptmask` using `loadmeasure` binary.
2. Prepare the input files to generate forecasts. These two files are necessary:
  - A datelist file contains the list of desired forecast start date in the format of YYYYMMDD (for example, 20101130 and as shown in [Example 3-1](#))
  - An input file to `PreGenerateForecast` binary (`pregen.xml`)

The forecast start dates need to be seven (7) days apart as shown in [Example 3-1](#).

### **Example 3-1 Datelist File**

```
20100101
20100108
20100114
20100120
```

The `pregen.xml` and `generate.xml` needs to have the right input for the desired forecast level and the desired path for the generate xml file input

3. Run `exportRDFtoAPCRO.ksh` to generate the output file for APC RO. This script performs the following steps:
  - a. For each date in the `dateList`:
    - Clear Approved Forecast and Approved Cumulative interval
    - Set forecast start date to the current date
    - Run `PreGenerateForecast` binary using the provided path of input file.
    - For each local domain of the RDF domain: Run `Generate` using the provided path of input file. The provided path of inputfile needs to be the same as what specified in the provided path of input file for `PreGenerateForecast`.
    - Call `exportRDFtoAPCROWeekly.ksh` to export the newly generated approved forecast, approved cumint and other outputs.
  - b. Combine all the output created by the previous step into one file with the name specified by the `-out` argument.

## Usage

The usage of exportRDFtoAPCRO.ksh is as follows:

### Script Name

exportRDFtoAPCRO.ksh

### Required Arguments

The following table lists the required arguments for the script, exportRDFtoAPCRO.ksh.

Argument	Description
-d	Path to the master domain of a RDF domain (for example, /user1/domain/RDF)
-dateList	A file that contains a list of forecast start date in format YYYYMMDD (for example, 20101130)
-PregenXML	Path to the pregenerateforecast input xml file (for example, /user1/domain/RDF/pergen.xml)
-GenXML	Path to the generate input xml file (for example, /user1/domain/RDF/generate.xml, produced by the pregenerateforecast binary)
-ForecastLevel	Forecast level number in format xx, (for example, 01, 06)
-out	Output file name (for example, ./user1/domain/RDF/output/toapcro.csv)

### Optional Arguments

The following table lists the optional arguments for the script, exportRDFtoAPCRO.ksh.

Argument	Description
-DOWProfile	The optional day of week profile measure name. (for example, apvp11)
-mask	The mask measure name (for example, RDFAPCROMsk01)
-maxProcesses	Specifies the maximum number of parallel processes (for example, 3). The default is 1.
-sundayIndex	Sunday's position (between 1 & 7) in the DOW hierarchy, The default is 2.
-switchOrder	Depending on your configuration, Prod hierarchy may be in front of or behind the LOC hierarchy, but in the output file the Prod ID is required to be always in front of Loc ID. So turn this switch either on or off if you see that the order in the output is incorrect. (for example, <b>false</b> since the default is <b>true</b> .)

## Integration Script

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

- The exportMeasure utility is used to export data in CSV (comma-separated values) format. This maintains the consistency of start and width attributes across different applications.
- Data exported from the source application is placed in the destination domain input directory.
- Export scripts must run before load scripts. They should be run in the batch window.
- 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 3–1 lists the integration scripts for RDF.

**Table 3–1 Integration Script**

Application	Script Name
PPO	exportRDFtoPPO.ksh
AIP	rdf_e_aip_appf.ksh rdf_e_aip_cumint.ksh
RMS	rdf_e_rms.ksh



---

---

## ODI Integration

This chapter describes the integration of APC-RPO, RPO, and RDF using Oracle Data Integrator (ODI). For information about script integration among these applications, see [Chapter 3, "Integration"](#).

### ODI Integration

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

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

### Measure Data Integration

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

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

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

`$RPAS_HOME/scripts/integration/ODI`

### APC-RPO to RPO Package

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

#### Data Mapping for APC-RPO to RPO Package

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

**Table 4–1 APC-RPO to RPO Data**

APC-RPO Expression	RPO Target Measure
exptelst	ol1gammaasp
anchorprice	ol1anchprc
minhistprice	ol1hisloprc
maxhistprice	ol1hishiprc

## APC-RPO to RDF Package

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

### Data Mapping for APC-RPO to RDF Package

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

**Table 4–2 APC-RPO to RDF Data**

APC-RPO Expression	RDF Target Measure
achprstr	rdfanchprc
exptelsstr	rdfgamma
price	rdfprice

## RDF to RPO Package

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

### Data Mapping for RDF to RPO Package

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

**Table 4–3 RDF to RPO Data**

RDF Expression	RPO Target Measure
appf01xb	dl1itbdsp

## RPO to RDF Package

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

### Data Mapping for RPO to RDF Package

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

**Table 4–4 RPO to RDF Data**

RPO Expression	RDF Target Measure
fappitpc	rdfprice

---

---

## Installation Consideration

This chapter describes the setup that must be done before building the RDF domain and the batch script that must be executed to build the domain.

### Installation Dependencies

RPAS and RDF must be installed before setting up and configuring RDF:

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

### RPAS Installation

The Java-based RPAS installation programs that are included with the installation package are used to install the server-side RPAS components on UNIX operating systems.

The RPAS installer performs the following functions:

- Installs the RPAS server
- Installs the Configuration Tools on the server  
On Windows, an InstallShield package is used to install the Configuration Tools.
- Defines the DomainDaemon port

### RPAS Client Installation

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

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

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

## RDF Installation

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

The RDF installer automates the following tasks:

- Installs the RDF mock install configuration
- Installs RDF plug-ins for the Configuration Tools
- Installs Language Translation files
- Creates a sample RDF domain

## RDF Taskflow for the RPAS Fusion Client

The RDF 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*.

## Environment Variables

In addition to the regular RPAS environment variables, including RPAS\_HOME, you need to set up the following environment variables and export them.

Environment Variable	Description	Use
RIDE_HOME	The location of the configuration tools.	Mandatory for Installation and Patching
PATH	The standard operating system path. Running the script \$RPAS_HOME/rpaslogin.ksh is the preferred method to update and export the PATH variable.	Mandatory
RDF_BATCH_TIMEOUT	This numeric value, set in seconds, controls the amount of time that the Batch Forecast workbook will wait for another Batch Forecast workbook process to finish. This allows several users in different local domains to start a Batch Forecast workbook simultaneously.	Optional

## Files Needed to Build the RDF Domain

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

- Standard RPAS hierarchy files
- Data files

## Standard RPAS Hierarchy Files

The following hierarchy files are needed:

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

---



---

**Note:** As with all standard RPAS hierarchies, these hierarchies are configurable. For information about configuring these hierarchies, see [Chapter 6, "Configuration Considerations"](#).

---



---

### Calendar (CLND) Hierarchy File

**File name:** clnd.csv.dat

**File format:** comma-separated values file

[Table 5–1](#) describes the fields in the file:

**Table 5–1** *Calendar Hierarchy Fields*

Field	Description
Day	Day ID
Day label	Day label
Week	Week ID
Week label	Week label
Mnth	Month ID
Mnth label	Month label
Qtrtr	Quarter ID
Qtrtr label	Quarter label
Fiscal Half	Fiscal Half ID
Fiscal Half label	Fiscal Half label
Year	Year ID
Year label	Year label
Day of Week	Day of Week ID
Day of Week label	Day of Week label
Day of Season	Day of Season ID
Day of Season label	Day of Season label
Week of Year	Week of Year ID
Week of Year label	Week of Year label
Week of Season	Week of Season ID
Week of Season label	Week of Season label

#### Example:

```
20050130,01/30/2005,w01_2005,01/30/2005,JAN_2005,January
2005,Q1_2005,Quarter 1 2005,H1_2005,2005 First Half, A2005, Year
2005, SAT, Saturday,DOS01,DOS 01,WY01, Week 01, WS01,WOS 01
```

**Product (PROD) Hierarchy File****File name:** prod.csv.dat**File format:** comma-separated values file

Table 5–2 describes the fields in the file:

**Table 5–2 Product Hierarchy Fields**

Field	Description
SKU	SKU ID
SKU label	SKU label
Parent	Parent ID
Parent label	Parent label
Grand Parent	Grand Parent ID
Grand Parent label	Grand Parent label
Subclass	Subclass ID
Subclass label	Subclass label
Cls	Class ID
Cls label	Class label
Dept	Department ID
Dept label	Department label
Group	Group
Group label	Group label
Division	Division ID
Division label	Division label
Supplier	Supplier ID
Supplier Label	Supplier label
Diff 1	Diff 1 ID
Diff 1 label	Diff 1 label
Parent Diff 1	Parent Diff 1ID
Parent Diff 1 label	Parent Diff 1 label
Grand Parent Diff 1	Grand Parent Diff 1 ID
Grand Parent Diff 1 label	Grand Parent Diff 1 label
Subclass Diff 1	Subclass Diff 1 ID
Subclass Diff 1 label	Subclass Diff 1 label
Class Diff 1	Class Diff 1 ID
Class Diff 1 label	Class Diff 1 label
Dept Diff 1	Dept Diff 1 ID
Dept Diff 1 label	Dept Diff 1 label

**Example:**

```
10000010,10000010Leather Loafer - Black 6 B, 10000010,
10000010Leather Loafer - Black 6 B, 10000009, 10000009Leather
```

Loafer, 122, 122Loafer, 1312, 1312Casual, 1310, 1310Footwear  
 Women's, 1300, Group 1, 1, All Product, 1000, Supplier 1,  
 10000010\_sml, 10000010Leather Loafer - Black 6 B Small,  
 10000009\_sml, 10000009Leather Loafer Small, 122\_sml, 122Loafer  
 Small, 1312\_sml,, 1312Casual\* Small, 1310\_sml,, 1310Footwear  
 Women's\* Small, \_sml, Small

### Location (LOC) Hierarchy File

**File name:** loc.csv.dat

**File format:** comma-separated values file

Table 5–3 describes the fields in the file:

**Table 5–3 Location Hierarchy Fields**

Field	Description
Str	Store ID
Str label	Store label
District	District ID
District label	District label
Regn	Region ID
Regn label	Region label
Area	Area ID
Area label	Area label
Chnl	Chain ID
Chnl label	Chain label
Company	Company ID
Company label	Company label
Store Format	Store Format ID
Store Format label	Store Format label
Store Class	Store Class
Store Class Label	Store Class Label

#### Example:

1000, New York City, 1000, US, 1000, North America, 1000, The  
 Americas, 1000, Bricks & Mortar, 100, JCB Trading Company, 4, 4,  
 A, A

## Required Data Files

The following data files are required:

- Sales history
- Promotion history (if using RDF causal)

**File format:** comma-separated values file

Table 5–4 lists examples of the required data files.

**Table 5–4 Examples of Required Data Files**

Required Data File	Example
Sales history	20090311,10000044,1000,8 20090415,10000044,1000,5
Promotion history (if using RDF causal)	20090311,10000044,1000,1 20090415,10000044,1000,1

## Optional Data Files

The following data files are optional:

- Like items
- Out of Stock Indicators
- Sister stores

**File format:** comma-separated values file

**Examples of the Out of Stock Indicators Data File:**

20090311,10000044,1000,1

20090415,10000044,1000,1

## Output from RDF to RMS and Retail Analytics

- Weekly forecasts and cumulative intervals
- SKU/Store/Week
- Daily forecasts and cumulative intervals
- SKU/Store/Day

---

---

## Configuration Considerations

RDF is a statistical forecasting solution that uses state-of-the-art modeling techniques to produce high quality forecasts with minimal human intervention. Forecasts produced by RDF enhance the retailer's supply-chain planning, allocation, and replenishment processes, which enables a profitable and customer-oriented approach to predicting and meeting product demand.

Forecast information is often required for items at the lowest levels in a hierarchy. Problems can arise when historic sales data for these items is too sparse and too noisy to identify clear selling patterns. In such cases, generating a reliable forecast requires aggregating sales data from a low level up to a higher level in the hierarchy. After a forecast is generated at the higher level, the resulting data can be allocated (spread) back down to the lower level. This is based on the lower level's relationship to the total. Before you can spread forecast data back down to a lower level, you should have an understanding of the relationship between the lower level and the higher level dimensions. Frequently, an additional forecast will be generated at the low level to help determine this relationship. This low level is called the final forecast level. Forecast data at this level might be sufficient to generate reliable percentage-to-whole information, but the actual forecast numbers will be more robust when they are generated at an aggregate level. This aggregate level from which forecast data is spread is referred to as the source forecast level.

Some high-volume items may possess sufficient sales data for robust forecast calculations directly at the final forecast level. In these cases, forecast data that is generated at an aggregate level and then spread down to lower levels can be compared to forecasts that are run directly at the low level. Comparing the two forecasts, each generated at a different hierarchy level, can be an invaluable forecast performance evaluation tool.

The RDF solution may include multiple final forecast levels. Forecast data must appear at some final level for the data to be approved and exported to other systems.

Using the RDF plug-In, final and source forecast levels are defined for the RDF solution.

---

---

**Note:** The ability to configure the RDF solution may be limited. This is based on your licensing agreement.

---

---

## Forecasting Calendar Hierarchy Requirement

With any RDF solution, configuration of the calendar hierarchy must always include a day dimension level name. There are no configuration requirements for the dimensions of the merchandise or location hierarchies.

## Forecasting Limitations Using the Partition Hierarchy

Any dimension along the partition hierarchy that is used as an intersection to forecast must be unique across all domains. This requirement especially applies to Alternate Hierarchies. For example, if the forecast level is supplier\str\week, the Supplier dimension cannot have a supplier position that exists in multiple domains. However, additional support for clean partitioning of Alternate Hierarchies is provided through the RDF Transformation programs used to integrate RMS foundation data for RDF.

## Causal Forecasting at Source Levels

RDF expects all promotional history to be pre-aggregated (externally or through custom RPAS rules) to each source level, when running causal forecasting at that level. It is possible to enable/disable promotions for each causal forecast level within the application.

## Loc Hierarchy Limitation

RDF expects that the location hierarchy is called loc.

## Forecasting Pre-Configuration Data Requirements

There are several parameters within the RDF configuration that may reference other measures that are configured external to the solution, specifically:

- Source Data
- Seasonal Profile
- Plan Data
- Spreading Profile

Prior to configuring an RDF solution, it is required that these measures already exist within the Project.

### Source Data

The RDF plug-in populates a pick-list with all non-Boolean and non-string measures that have been created in the Project.

### Plan Data

If the Plan Data that will be used to support Bayesian forecasting is being defined within another solution, this measure should already exist. The entry of this parameter is not required within the configuration, and it can be entered in the resulting domains.

### Spreading Profiles and Seasonal Profiles

If Curve will be used to produce Spreading Profiles or Seasonal Profiles to support your Forecasting solution, these profiles should already have been configured in the

Curve solution. If these profiles are being defined external to Curve, these measures should already exist within the Project.

## Registering the RdfFunctions Library

Prior to configuring the RDF Solution, register the RdfFunctions library to support proper validation of the RDF-specific rules:

Open the Function Library Manager and add RdfFunctions.

---



---

**Note:** If Promote is implemented, the following rules will display as invalid; however these should be ignored:

- Rule: PREF\_PIHolder
  - RuleGroup: PREF\_place
  - Rule Group: PRMA\_place
  - Rule Group: PRPL\_place
- 
- 

## Editing Forecast Level Parameters

---



---

**Note:** For more information on Source Level Forecasting, see the *Oracle Retail Demand Forecasting User Guide*.

---



---

Table 6–1 lists and describes the forecast level parameters:

**Table 6–1 Forecast Level Parameters**

Forecast Level Parameters	Description
Level Name	The Level Name is the system-assigned level number when a forecast level is created. This is a read-only parameter.
Level Label	The Level Label is the level description that will be viewed by the user once the domain is created. See " <a href="#">Level Labels</a> " for additional information.
Intersection	The Intersection is the hierarchy dimensions that define the forecasting level.
Default Source Level	Assigned only at the Final level, the Default Source Level is the primary level at which the aggregate, more robust forecast is run. The desired Source Level must first be created within the RDF configuration for it to be a selection in the pick-list. For more information on Source Level Forecasting, refer to the <i>Oracle Retail Demand Forecasting User Guide</i> .  If no source level is required, the final level should be selected.
Source Data	Assigned only at the Final level, the Source Data is the measure to be used as the input data (for example, POS) for the generation of forecasts. The values in this pick-list are populated with all non-string and non-Boolean type measures that are configured in the Project.
Periodicity	Periodicity is the number of periods within the Calendar dimension, which are defined in the forecast level intersection.  For example, if an intersection is defined at Week/item/store the Periodicity value will be 52 (since there are 52 weeks within a year).

**Table 6–1 (Cont.) Forecast Level Parameters**

Forecast Level Parameters	Description
Forecast Method	The Forecast Method window displays all forecast generation methods that may be defined for a forecast level. The Default Forecast Method is also determined here.  See " <a href="#">Forecast Methods</a> " for a list of Forecast Methods that may be selected. See the <i>Oracle Retail Demand Forecasting User Guide</i> for more information on each method.
Plan Data	Assigned only at the final level, Plan Data (sales plans) provide details of the anticipated shape and scale of an item's selling pattern. This information is required when Bayesian forecasting is used as a Forecast Method. The value in this parameter is a measure name.
Seasonal Profile	A seasonal profile provides details of the anticipated seasonality of an item's selling pattern. The seasonal profile is required in conjunction with the Profile-based Forecast Method. The seasonal profile can be generated or loaded, depending on your configuration. The value in this parameter is a measure name.
Spreading Profile	Assigned only at the source forecasting level, the Spreading Profile is used to spread source level forecasts down to the final forecast level. The value in this parameter is a measure name, a profile level name, or any combination of these separated by commas. <ul style="list-style-type: none"> <li>▪ If Curve is used to dynamically generate the spreading ratios, this parameter should be populated with the final profile level name (profile number) configured. For example: 01 (this is profile level 01).</li> <li>▪ If Curve is used to generate the static (manually approved) spreading ratios, this parameter should be populated with the Approved Profile measure. For example: apvp11 (this is the Approved Profile for Curve level 11).</li> </ul>

## Level Labels

The Level Label is the level description that will be viewed by the user once the domain is created.

- Level Labels may not exceed 40 characters.
- It is recommended, but not required, that Level Labels include the Level Name (the system-assigned level number).

Within the Forecast Administration workbook, the Default Source Level may be edited. This pick-list is populated with the Level Name for all levels that are associated with a final level. Since this value can also be specified within this configuration, this recommendation may not be necessary if changes to the Default Source Level are not expected within the application.

- RPAS automatically places parentheses ( ) around Forecast Level Labels.

The configuration specialist should not include these in the level label configuration or the installer will fail.

Correct Example	Incorrect Example
1- itm/str/week - Final	(1:itm/str/week - Final)

- A hyphen (-) should not be used before or after the Forecast Level Label.

Correct Example	Incorrect Example
1-itm/str/week - Final	-1:itm/str/week - Final-

- A colon (:) should not be used at all in the Level Label.

Incorrect Example
1: itm/str/week-

## Forecast Methods

The following is a list of Forecast Methods that may be selected. See the *Oracle Retail Demand Forecasting User Guide* for more information on each method.

- No Forecast
- Average
- Simple
- Intermittent
- Simple/Intermittent
- Trend
- Additive Seasonal
- Multiplicative Seasonal
- Seasonal
- AutoES
- Causal

---



---

**Note:** See "[About Causal Forecasting](#)" for special conditions for Causal methods.

---



---

- Bayesian
- Profile-based
- LoadPlan
- Copy

### About Causal Forecasting

The Causal method should be selected as a valid method only for levels in which causal forecasting will be used.

When enabling Causal as a valid forecast method for a source level, note that RDF Promotion variables need to be provided at the same dimension along the product and location hierarchies as the forecast level for which Causal forecasting is executed (Final or Source). RDF Causal does not support aggregation of promotion variables along any hierarchies other than Calendar (Clnd). Aggregation of promotion variables along product and/or location hierarchies needs to be handled externally through configuration. Aggregation along the calendar hierarchy is support by RDF Causal, using specified aggregation and spread profiles. Refer to the *Oracle Retail Demand Forecasting User Guide* for details.

## Autogenerating Hierarchies, Measures, Rules and Workbook Templates

The following is the process to autogenerate the hierarchies, measures, rules, and workbook templates that are required by RDF to support the forecasting configuration entered in the RDF plug-in:

The system automatically generates the following:

Item	Description
Hierarchies	The DATA hierarchy will be updated with the flvl, fbrt and fmtr dimensions.
Measures	All measures necessary to support the base RDF solution will be created.
Rules	All Rule Sets, Rule Groups, and Rules to support the base RDF solution will be created.
Workbook Templates	All pre-defined workbook templates to support the base RDF solution will be created.

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

### Deleting a Forecast Level

Deleting a forecast level will cause the system-assigned enumerated values in the Level Name to renumber such that levels are in consecutive order starting with forecast level 01. Deleting a forecast level may impact any solution configuration that uses a specific level.

If the domain using the configuration has previously been installed, there is potential to lose data associated to a level that has been deleted or renumbered.

### Configuring the Cloning Administration Workbook

The Product/Location Cloning Administration workbook allows users to specify clone products by a configurable dimension in the location hierarchy and clone stores by a configurable dimension in the product hierarchy. For example, users can specify a different clone SKU for a different region.

### Editing the RDF GA Configuration

The autogeneration process creates hierarchies, measures, rules, and workbook templates that are required to support the essential RDF functionality. This base configuration is referred to as the GA Configuration. Certain changes to the GA Configuration are allowed. Once edits to the GA Configuration are made and the autogeneration process occurs again, valid changes to the configuration will be preserved. There is nothing in the RPAS Configuration Tools to prevent invalid changes from being made.

The following table outlines acceptable changes and restrictions.

Editable Item	Description
RDF Solution Extension Name	The name assigned to the resulting RDF solution after autogeneration occurs cannot be edited.
Major and Minor Components	Additional major and minor components may be added to the RDF GA Configuration. The major and minor components that are part of the GA Configuration may not be edited. This restriction also applies to measure names and measure labels.
Rules	Additional Rule Sets, Rule Groups, and Rules may be added to the RDF GA Configuration. This includes support for adding new rules to existing GA configuration rule groups. It is recommended that new rules added to the GA configuration rule groups include cust (represents Custom) in the rule name. This allows for easy identification of Rules that are not part of the GA Configuration. Rule sets, rule groups, and rules that are part of the GA Configuration may not be renamed. Existing rules that are part of the GA Configuration may not be modified in any way.
Workbook Templates	Additional workbook templates may be added to the RDF GA Configuration. New measures and rules may also be added to the GA configuration workbook templates. This is done by adding new major and minor components, and adding new Rules to existing rule groups in the GA Configuration.

## RDF Non-modifiable Hierarchies

The names of the hierarchies in this section cannot be changed.

**Table 6–2 Non-Modified Hierarchies**

Hierarchy Name	Hierarchy Label
DATA	Data Hierarchy
CLSH	Cluster
GRCH	Grade configurations
PRMH	Promotions
CSLH	Causal levels

### Calendar (CLND)

The Calendar hierarchy represents time in all RPAS solutions. It is a required hierarchy and must have a dimension named day (DAY).

### Product (PROD)

The Product or Merchandise hierarchy represents the retailer's merchandise (that is, merchandise that the retailer retails through its retail channels).

### Location (LOC)

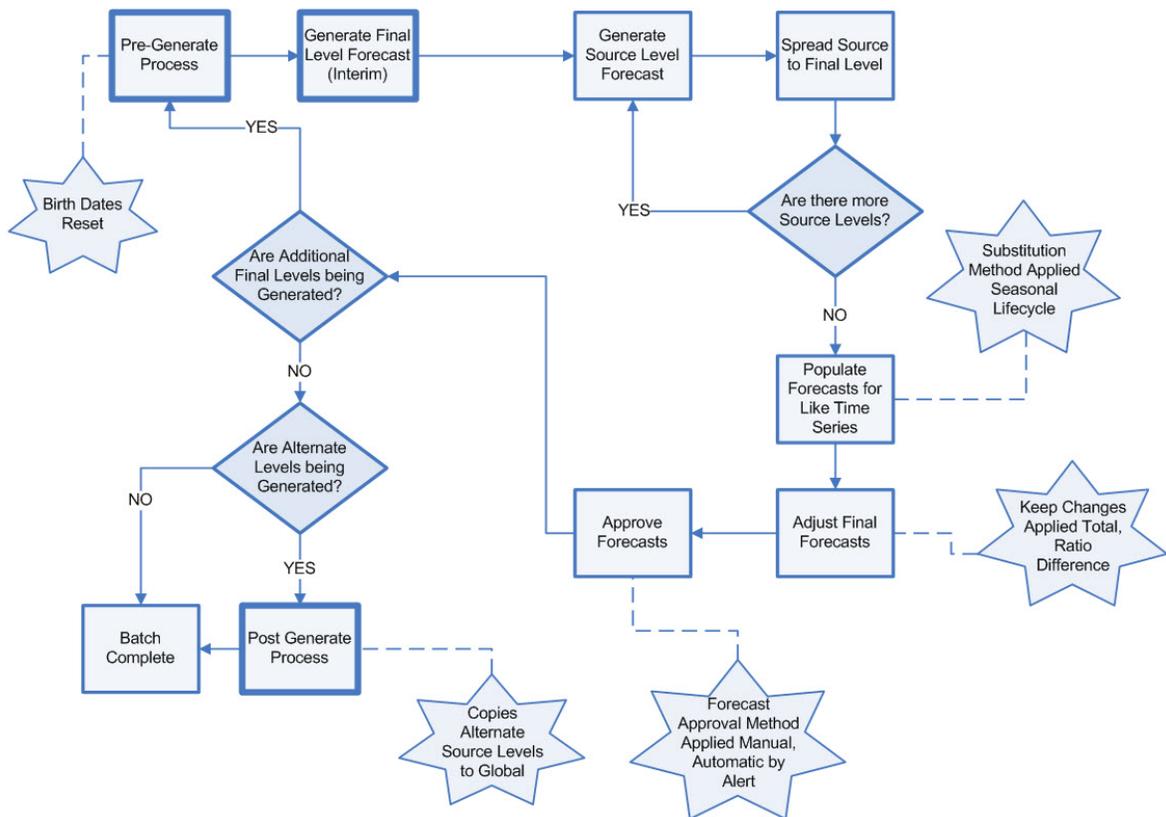
The Location hierarchy represents the retailer's retail locations and their roll-ups.



## Batch Processing

Figure 7-1 provides a high-level overview of the batch forecast process.

**Figure 7-1 Overview of the Batch Forecast Generation Process**



## About RDF Batch Scripts

Batch scripts, which are configured externally, are required by RDF to execute processing before, during and after forecasting. This section outlines a sequence of common batch scripts that may need to be configured during RDF implementation to achieve desired processing to meet your business needs.

### RDF Binaries

This section lists all RDF binaries with details and usage information

#### 1. Hierarchy Loads

Use the RPAS utility loadHier to load standard hierarchy files into RDF. The RPAS utilities positionBufferMgr and reconfigGlobalDomainPartitions may also need to be called by the loadHier script.

Note that RPAS supports centralized hierarchy loads, which means that this script can be executed from the master domain, in a global domain environment.

---

---

**Note:** If user-defined hierarchies are present, then always run reshape arrays before continuing.

---

---

#### 2. Measure Loads

Use the RPAS utility loadMeasure to load sales history, out of stock information, promotion variables and other necessary data into RDF.

Note that RPAS now supports centralized measure loads, which means that this script can be executed from the master domain, in a global domain environment.

#### 3. Preprocessing

Performs any preprocessing calculations needed for forecasting. This script would invoke the preprocessing rule group that would be configured in the RDF configuration. Preprocessing performs any necessary scrubbing of historic data before forecast generation.

This script must be executed from local domains, if the preprocessing calculations involve RHS and LHS measures that are non-HBI, as would typically be the case for preprocessing calculations. Parallel execution along local domains is possible.

#### 4. Forecast Generation

In order to generate forecasts, execute the following RDF files:

- PreGenerateForecast

This file is executed from a master domain and performed before running Generate.

- Generate

This file is run from local domains in a global domain environment.

#### 5. UpdateFnhbiRdf

This is an optional script, which is needed only if an alternate hierarchy dimension from the Product hierarchy is used as a dimension in a forecast level.

This script needs to be run from the master domain, unless it is known that only one local domain has forecast data. Then calling this script from that local domain can save some time.

---



---

**Note:** If more than one local domain may have forecast data, then this script must be called from the master domain.

---



---

## 6. Alert Manager

Use the RPAS utility `alertmgr` to evaluate alert conditions specified in the RDF configuration.

With RPAS 12.0.6 and beyond, it is now possible to run `alertmgr` from local domains, followed by a final synchronizing run from the master domain, to synchronize alert hit counts at the master domain level. This can be achieved by running `alertmgr -findAlerts` from the local domains. After running `alertmgr -findAlerts` from the local domains, run `alertmgr-sumAlerts` from the master domain. The `alertmgr -findAlerts` process is more calculation intensive, but it can be performed in parallel at the local domain level.

## 7. Export Forecasts

Use the RPAS utility `exportData` to export RDF forecasts from RDF for use by external systems. Users typically export the Approved forecasts from RDF. Also, please refer to the RDF integration scripts (`rdf_e_rms.sh`) packaged with RPAS.

This script is run from local domains.

## 8. Autoworkbook Build

This script performs any necessary automated workbook builds as set up by the user. Automated workbook builds are set up by users to automate the workbook build process, so that they do not have to make the same wizard selections each time the workbook is built, and they do not have to wait for workbooks to build. The underlying RPAS utility used is `wbatch`. This needs to be run from the local domains.

# PreGenerateForecast

Used in a global domain or simple domain environment, `PreGenerateForecast` is an RDF executable that registers all measures with a birth date prior to forecast generation using `generate`. The first time `PreGenerateForecast` is run for a level, it registers the appropriate token measures for that level. If a global domain environment is implemented, `PreGenerateForecast` may be run against the Master or a Local domain. At either level, the necessary measures to produce the batch forecast will be registered across all domains.

`PreGenerateForecast` requires an input file in the form of an XML. The XML is configured with the following values:

- **FinalLevel**

The Final Level Number that will be used to generate the forecast.

- **OutputFile**

The name of the resulting file located at the root of the domain after `PreGenerateForecast` is run. The `OutputFile` includes the values set for `FinalLevel` and `Override` in addition to the birth date. This date is the Forecast Generation Date, and it is passed to the domains when `generate` is run.

The date is produced in the following format: `yyyymmddHhhMmm` (Example: `20050327H13M36`). When this birth date is selected in the Forecast Approval wizard, it will be viewed as: `(03/27/2005 13:36)`.

- **Override**

A **true** or **false** value. When generate is passed a **true** value, the Next Run Date is ignored, and the batch forecast uses today's date as the Next Run Date; and the batch is executed. When generate is passed a **false** value, the batch forecast will run if the Next Run Date is the same as today's date.

---

---

**Note:** When the Run Batch template is used to generate the batch forecast, PreGenerateForecast is run automatically. If a global domain environment is implemented, forecasts produced across Local domains using Run Batch cannot be aggregated in the Master domain because they do not share the same Forecast Generation Date.

---

---

## PreGenerateForecast Usage

PreGenerateForecast -InputFile filename

InputFile is required.

The input file should be an XML file that looks like this:

```
<Parameters>
  <Parameter>
    <Key>FinalLevel</Key>
    <Value>1</Value>
  </Parameter>

  <Parameter>
    <Key>OutputFile</Key>
    <Value>MyOutput.xml</Value>
  </Parameter>
  <Parameter>
    <Key>Override</Key>
    <Value>>true</Value>
  </Parameter>
</Parameters>
```

FinalLevel and OutputFile are required parameters of the XML file.

Override is an optional parameter of the XML file (default is **false**).

Other parameters may be included in the input XML file. They will be passed through to the output XML file.

Return codes:

- 0 - Success (either ran pre-generate or did not need to run)
- 1 - Bad input
- 2 - Failure

To set the level of information provided, use -loglevel with values of: all, profile, debug, information, warning, error, or none. To disable timestamp header use -noheader.

## generate

Used to produce the batch forecast, generate is an RDF executable. This executable requires as an input, the OutputFile resulting from PreGenerateForecast and is called generate.xml.

This binary runs RDF's batch process. Generate can take two optional inputs: level and override.

## Usage

```
generate -InputFile Filename
```

The following parameters setting are included in the input file:

- birth
- startdate
- finallevel
- override

The override input must be **true** or **false**. The defaulted value is **false** if this option is not included in the input file. When override is **false**, generate will only start the batch process if current time is later than the next run date in the domain. When the override is **true**, generate will start the batch forecast regardless of the next run date.

The generate binary invokes code in the BatchForecast library to execute the batch process.

finalLevel and birth are required parameters of the XML file. override (**false**) and StartDate (Default Forecast Start Date) are optional parameters of the XML file (defaults in parentheses).

## Return Codes

The return codes include:

- 0—Success (either ran generate or did not need to run)
- 1—Bad input
- 2—Failure

To set the level of information provided, use -loglevel with values of:

- all
- profile
- debug
- information
- warning
- error
- none

To disable timestamp header use -noheader.

The input file should be an XML file that looks similar to the following example:

```
<Parameters>
  <Parameter>
    <Key>Birth</Key>
    <Value>20041027H11M52</Value>
  </Parameter>
  <Parameter>
    <Key>StartDate</Key>
    <Value>20041027</Value>
  </Parameter>
  <Parameter>
    <Key>FinalLevel</Key>
    <Value>1</Value>
  </Parameter>
  <Parameter>
    <Key>Override</Key>
    <Value>>true</Value>
  </Parameter>
</Parameters>
```

## RDFvalidate

RDFvalidate automatically executes during the domain install, and it can also be run at any time against a Master or one subdomain. If run against the Master Domain, it checks the master and all subdomains. If run against a subdomain, it checks the Master and only the subdomain (not all other subdomains). This function verifies that:

- If there is a partition dimension, it must be along the product hierarchy.
- Domains are cleanly partitioned, this means that for the partition dimension, there exists only one position in each local domain, whether partitioning along the main or an alternate (or branch) product hierarchy.
- All data, measures, and levels are defined properly based on the partition dimension.
- Causal parameters are properly defined based on final, source, and causal levels.

## Usage

```
rdfvalidate -d pathToDomain
```

To get this usage text, use `-?`, `-help`, or `-usage`. To get the version of this utility, use `-version`. To set the level of information provided, use `-loglevel` with values of: `all`, `profile`, `debug`, `information`, `warning`, `error`, or `none`. To disable timestamp header use `-noheader`.

## RDF Validation

Table 7–1 displays the validation performed internally by the plug-in and the RDFvalidate utility.

**Table 7–1 Internal Validation Performed by the Plug-in and RDFvalidate utility**

#	Validation Area	Steps
1.	Hierarchies and Dimensions	<p>a. Verify day dimension exists on calendar hierarchy</p> <p>b. If there is a partition dimension, it must be along the product hierarchy.</p>
2.	For Final Levels	<p>a. Intersection (fintxldb)</p> <ul style="list-style-type: none"> <li>▪ Cannot be blank</li> <li>▪ Must be at or below all source level intersections</li> <li>▪ Must be at or below the partition dimension on the partition branch</li> </ul> <p>b. Seasonal profile (seasprofldb) can be either:</p> <ul style="list-style-type: none"> <li>▪ Blank</li> <li>▪ Measure name (only one) <ul style="list-style-type: none"> <li>▪ Must be valid measure</li> <li>▪ Should be of type real</li> <li>▪ Measure intersection must be equal to the level intersection</li> </ul> </li> </ul> <p>c. Source data (datasrcldb) must be a measure name (only one)</p> <ul style="list-style-type: none"> <li>▪ Must be a valid measure</li> <li>▪ Should be of type real</li> <li>▪ Measure intersection must be at or below the final level intersection</li> </ul> <p>d. Plan data (rfplanldb) must be either:</p> <ul style="list-style-type: none"> <li>▪ Blank</li> <li>▪ Measure name (only one) <ul style="list-style-type: none"> <li>▪ Must be valid measure</li> <li>▪ Should be of type real</li> <li>▪ Measure intersection must be equal to the final level intersection</li> </ul> </li> </ul>

**Table 7–1 (Cont.) Internal Validation Performed by the Plug-in and RDFvalidate utility**

#	Validation Area	Steps
3.	For Source Levels:	<p><b>a. Intersection (fintxlb)</b></p> <ul style="list-style-type: none"> <li>■ Cannot be blank</li> <li>■ Must be at or above final level intersection</li> <li>■ Must contain a dimension from the partition hierarchy</li> <li>■ Must be either:                             <ul style="list-style-type: none"> <li>■ At or below the partition dimension on the partition branch.</li> <li>■ On a branch of the partition hierarchy. If on a branch of the partition hierarchy, also check if domains are cleanly partitioned (executable only). This means for the branched dimension on the partition hierarchy, each position for that dimension can exist in only one sub-domain.</li> </ul> </li> </ul> <hr/> <p><b>b. Seasonal profile (seasprofxb) can be either:</b></p> <ul style="list-style-type: none"> <li>■ Blank</li> <li>■ Measure name (only one)                             <ul style="list-style-type: none"> <li>■ Must be valid measure</li> <li>■ Should be of type real</li> <li>■ Measure intersection must be equal to the level intersection</li> </ul> </li> </ul> <hr/> <p><b>c. Spreading profiles (sprdprofxb)</b></p> <ul style="list-style-type: none"> <li>■ Can only be blank if source level intersection equals final level intersection</li> <li>■ Must be comma-separated list of curve levels and measure names (can be mixed)                             <ul style="list-style-type: none"> <li>■ If curve level, must be a valid curve level (final profile)</li> <li>■ If measure:                                     <ul style="list-style-type: none"> <li>■ Must be a valid measure</li> <li>■ Should be of type real</li> <li>■ Measure intersection must be at or above final level</li> </ul> </li> </ul> </li> </ul>

## Executable Only

Table 7-2 displays the validation performed internally by the RDFvalidate utility.

**Table 7-2 Executable Only**

#	Executable Only	Steps
1	Domains are Cleanly Partitioned	a. Verify that there is only one partition dimension per subdomain.
2	For Final and Source Levels	<p>a. Causal Aggregation Profile (aggxlb) values should be either:</p> <ul style="list-style-type: none"> <li>■ Blank</li> <li>■ Measure name (one only) <ul style="list-style-type: none"> <li>■ Should be a valid measure</li> <li>■ Should be of type real</li> <li>■ The intersection of the measure must be at or above final level</li> </ul> </li> </ul> <p>b. Causal Calculation Intersection (calcintxlb) values should be either:</p> <ul style="list-style-type: none"> <li>■ Blank</li> <li>■ Intersection</li> </ul> <p>Must be valid intersection</p> <ul style="list-style-type: none"> <li>■ Must contain the calendar dimension</li> <li>■ Must be at or above level intersection</li> </ul> <p>c. Causal Data Source (calcdsrcxlb) values should be either:</p> <ul style="list-style-type: none"> <li>■ Blank</li> <li>■ Measure name (one only) <ul style="list-style-type: none"> <li>■ Should be a valid measure</li> <li>■ Should be of type real</li> <li>■ The intersection of the measure must be at or above level intersection</li> </ul> </li> </ul> <p>d. CausalHigher Intersection (cslhint) values should be either:</p> <ul style="list-style-type: none"> <li>■ Blank</li> <li>■ Intersection <ul style="list-style-type: none"> <li>■ Must be valid intersection</li> <li>■ Must not contain the calendar dimension</li> <li>■ Must contain a dimension from the partition hierarchy.</li> <li>■ Must be at or above level intersection</li> <li>■ Must be either: <ul style="list-style-type: none"> <li>■ At or below the partition dimension on the partition branch.</li> <li>■ On a branch of the partition</li> </ul> </li> </ul> </li> </ul>
<p><b>Note:</b> If on a branch of the partition hierarchy, also check if domains are cleanly partitioned (executable only). This means that for the branched dimension on the partition hierarchy, each position for that dimension can exist in only one sub-domain.</p>		

**Table 7–2 (Cont.) Executable Only**

#	Executable Only	Steps
2. (continued)	For Final and Source Levels (continued)	<p>e. Causal Spread Profile (spreadlxb) values should be either:</p> <ul style="list-style-type: none"> <li>■ Blank</li> <li>■ Measure name (one only) <ul style="list-style-type: none"> <li>■ Should be a valid measure</li> <li>■ Should be of type real</li> <li>■ The intersection of the measure must be at or above final level</li> </ul> </li> </ul> <p>f. Deseasonalized Demand Array (ddemandlxb) values should be either:</p> <ul style="list-style-type: none"> <li>■ Blank</li> <li>■ Measure name (one only) <ul style="list-style-type: none"> <li>■ Should be a valid measure</li> <li>■ Should be of type real</li> <li>■ The intersection of the measure must be the level intersection less the calendar dimension</li> </ul> </li> </ul>
3.	For Final Levels only	<p>a. Default History Start Date (defhstdt) values should be either:</p> <ul style="list-style-type: none"> <li>■ Blank</li> <li>■ A date within the calendar</li> </ul> <p>b. Forecast Start Date (dfxlb) values should be either:</p> <ul style="list-style-type: none"> <li>■ Blank</li> <li>■ A date within the calendar</li> </ul>

### Promote Validation

Plug-in and Executable

1. Hierarchies and Dimensions:  
Check whether or not PTYP, FLVL, and PROM exist in Data Hierarchy. If not, create them.
2. Promotion Names:  
Check if promotion names have 1 to 4 characters.
3. Causal levels must be at or below the partition dimension on the partition branch.

## UpdateFnhbiRdf

UpdateFnhbiRdf is required after Generate is executed if an alternate hierarchy dimension from the Product hierarchy is used as a dimension in a forecast level. It performs the following functionality:

- Checks that certain measures are cleanly partitioned
- Copies corresponding cells (based on the partition) from each sub-domain to the master domain
- Runs automatically with the Run Batch wizard
- After ensuring that the FNHBI (Forced non-Higher Based Intersections) measures are cleanly partitioned, UpdateFnhbiRdf copies corresponding cells (based on the partition dimension) from each sub-domain into the master domain

## Usage

```
UpdateFnhbiRdf -d pathToDomain -InputFile filename
```

To get this usage text, use  `-?`,  `-help`, or  `-usage`. To get the version of this utility, use  `-version`. To set the level of information provided, use  `-loglevel` with values of:  `all`,  `profile`,  `debug`,  `information`,  `warning`,  `error`, or  `none`. To disable timestamp header, use  `-noheader`.

The  `InputFile` format expected is as printed by the usage information. The timestamp or the birth key will have to be the same as the one output by  `pregenerateForecast`, that is used by  `generate`.



---

---

## AutoSource

The AutoSource binary may be used to determine the optimal source level for a product/location. For the final level specified, AutoSource produces a forecast using each source level. The source level that produces the best PAE (Percent Absolute Error) for a time series is selected as the Optimal Source Level. The AutoSource results may be accessed by the user through the Forecast Maintenance workbook. If the Optimal Source Level is to be used for a product/location, the **Use Optimal Source** parameter should be set to **true**.

The AutoSource binary invokes code in the BatchForecast library to execute the batch process. AutoSource can take four inputs: mode (required), finallevels (required), today, and timelimit (required).

The AutoSource binary does the following:

- Provides a starting Source Level recommendation for new forecasting customers. The recommended Source Level can be applied to the Final Level, which would allow the user to be focused on other tuning activities.
- Is helpful for existing customers that are starting to forecast new businesses. AutoSource can be included as an activity in the customer's forecasting roll-out process.
- Is useful for merchandise groups that have shifting demand patterns due to business or market changes such as pricing and marketing strategy changes, or product realignment.

AutoSource uses the forecast horizon to compute the PAE (Percent Absolute Error). If the forecast horizon is changed from the default of 13 weeks, AutoSource will start forecasting that number of weeks back. For example, if you have a forecast horizon of 52 weeks, AutoSource will start its analysis 53 weeks before today. This approach can disallow Winters and Seasonal models if sufficient calendar is not available. If the forecast horizon is 52 weeks, you should have at least 3.5 years of history for AutoSource to be able to perform all of its analysis.

Unlike Generate, there is no interim forecast calculation in AutoSource. Instead, AutoSource attempts to generate an AutoES result at the final level, then uses that result to perform the source level spreads.

---

---

**Note:** If the time series data is dense enough at the final level, the spread will not be based on a linear contribution to the source. It will not maintain the source shape, and it will make recommendations based on such spreads.

---

---

AutoSource makes an initial recommendation to all the product/location combinations with sufficient data to perform analysis. Subsequent PAE calculation and comparison only occurs to these product/location combinations. The product/location combination without sufficient data (total sales = 0 during history region or total sales = 0 during forecast evaluation region) will not get any recommendation.

## Inputs to AutoSource Binary

AutoSource is invoked from a script or the command line. The binary inputs are detailed in [Table 8-1](#):

**Table 8-1 AutoSource Binary Input Descriptions**

Binary Inputs	Description	Example	
-d	Relative or absolute path to domain	-d C:\domains RDF	
-mode	Includes the following options:		
	RESTART	Resetting measures, such that the next run starts without prior information. This option does not actually kick off any source level optimization run. Use this option when a clean run is desired, and then run AutoSource with one of the following next modes.	-mode RESTART
	CYCLE	If AutoSource doesn't complete an optimization run due to the time limit, the next time it is run it picks up where it left last time.  For instance, if there are 10 source levels and during one run AutoSource only evaluated 3 source levels, then the next time it runs it optimizes source levels 4 and up.  CYCLE without a time limit will never finish. Once the last source level was evaluated, AutoSource starts with the first level again.	-mode CYCLE
	ONCEONLY	AutoSource completes the run or stops when the time limit is up	-mode ONCEONLY
-flvlist	A list of the final forecast levels to be optimized	-flvlist 1	
Optional Binary Inputs	Description	Example	
[-today]	Specifies the date when AutoSource stops the evaluation of the forecast error. The evaluation starts at the date given by today minus the number of periods specified in the forecast length. Hence the time interval over which AutoSource evaluates the forecast error is [today - forecast length, today]. The date should be in the RPAS format stored in the dim_day array.	-today D20010101	
[-timelimit]	Time, in minutes allowed AutoSource to run. If not specified, there is no time limit. Time limit is not an option if the runmode is ONCEONLY	-timelimit 10,000	
[-preserveTemp]	This is a flag indicating if temporary information should be deleted.  If not specified, the temporary information is deleted.	-preserveTemp	

### Example 1:

Autosource -mode RESTART -flvllist 1,6 -today DAY20050101 -timelimit 10

Autosource -mode ONCEONLY -flvllist 1,6 -today DAY20050101 -timelimit 10

**Example 2:**

Autosource -mode CYCLE -flvllist 1,6 -today DAY20050101 -timelimit 10

If only running AutoSource periodically, then use the RESTART and ONCEONLY modes. If the run exceeds the time limit during a RESTART run, then ONCEONLY should be run. If you want to start from the beginning, RESTART and ONCEONLY should be run again.

If AutoSource is scheduled as part of the daily cron job, use CYCLE. CYCLE runs RESTART and then ONCEONLY consecutively.

Refer to the Oracle Retail Demand Forecasting User Guide for specifics pertaining to the Forecast Maintenance Workbook and picking optimal levels.

---



---

**Note:** For item/stores that are new or highly seasonal, AutoSource may not return the best recommendation since new items may not have an adequate sales history length and highly seasonal items may only sell for a short period during the year. For these items, you should not set the AutoSource recommendation as default at the final level. Only use AutoSource recommendations for item/stores that have an adequate sales history.

---



---

## AutoSource Measures

The following AutoSource measures are available in the Forecast Maintenance workbook.

### Optimal Source Levels

Displayed only at final levels, a value will be populated in this field if AutoSource has been run on the final level. The AutoSource executable evaluates all levels associated to a final level and returns the Source Level that yields the optimal forecast results or lowest error.

### Pick Optimal Level

Set only at final levels, a check mark in this field indicates that the batch forecast should use the Optimal Source Level selected by AutoSource.

The final level measure Optimal Source Levels is used for reference. The RDF user can view the optimal Source Level that was determined by AutoSource. This Source Level was chosen by generating forecasts at all Source Levels and determining the lowest forecast error (PAE) at the final level.

If the user would like to use the Optimal Source Level during forecast generation they can set the Pick Optimal Level Boolean measure to **true**.

If Pick Optimal Level is set to **true**, when forecast generation is run, the optimal Source Level will be used. The Forecast Method set at the optimal Source Level and the additional associated forecast parameters will also be used.

**Usage**

autosource -d pathToDomain -mode RESTART/ONCEONLY/CYCLE -flvllist lvlx,lvly

[-today] todayString(the same format as YYYYMMDD)

[-timelimit] minutes [-preserveTemp]

To get this usage text, use `-?`, `-help`, or `-usage`.

To get the version of this utility, use `-version`.

To set the level of information provided, use `-loglevel` with values of: all, profile, debug, information, warning, error, or none.

To disable the timestamp header, use `-noheader`.

The mode input must be one of RESTART, CYCLE, or ONCEONLY.

The flvllist must be a comma separated list of final levels.

The today input must be the same format as YYYYMMDD.

The timelimit is in minutes.

- RESTART: This mode initializes the system in preparation for a new Autosource batch process.
- ONCEONLY: This mode will run the Autosource batch process until it completes or until the timelimit has been reached (whichever comes first).

---

---

**Note:** In order to run in ONCEONLY mode, RESTART mode has to be run first.

---

---

- CYCLE: This mode will continuously run the Autosource batch process by first running the RESTART mode, and then running ONCEONLY. The CYCLE mode allows the Autosource batch process to always use the latest data in determining the optimal source level for a prod/loc.

**Example 1:**

```
Autosource -mode RESTART -flvllist 1,6 -today DAY20050101 -timelimit 10
```

```
Autosource -mode ONCEONLY -flvllist 1,6 -today DAY20050101 -timelimit 10
```

**Example 2:**

```
Autosource -mode CYCLE -flvllist 1,6 -today DAY20050101 -timelimit 10
```

If AutoSource is to only run periodically, the RESTART and ONCEONLY modes should be used. If the run exceeds the time limit, then ONCEONLY should be run again to pick up where it left off. If you want to start from the beginning again, RESTART and ONCEONLY should be run.

If AutoSource is to be scheduled as part of the daily cron job, CYCLE should be used. CYCLE will run RESTART then ONCEONLY consecutively.

---

---

## Forecast Approval Alerts

This chapter describes the Forecast Approval alerts.

### Alerts

Alerts can be configured through the RPAS Configuration Tools or can be manually registered in the domain. The alert expressions require familiarity with the RPAS rule functions. Registering an alert with the alert category of FORECAST\_APPROVAL allows RDF to use the alert expression during the batch forecasting process to determine if a time series is automatically approved. When this category of alert is registered, the pick lists for Default Approval Method (in Forecast Administration) and the Approval Method Override (in Forecast Maintenance) are updated to include the label of the alert. The user can select the alert for any product/location.

The following is an example of Forecast Approval Alert configuration using the example domain that is provided in the release package.

### Step 0

#### **Build Global Domain**

Using the Mock Install Configuration, build the global domain environment.

### Step 1 (Option 1)

#### **Run PreGenerateForecast or Generate**

If using a pristine global domain or simple domain environment, token measures have yet to be registered in the domain(s). Since you do not know the specific birth date at configuration time, token measures allow for measures with birth dates (a time stamp applied during the batch) to be evaluated. The token measure that we are using in this example is System Forecast for level 1 (sf01). The registration of the token measures can be accomplished by running PreGenerateForecast (in a global domain environment) or Generate (in a simple domain environment). This removes the need to manually execute `regTokenMeasure`.

### Step 1 (Option 2)

#### **Use regTokenMeasure to Manually Register Any Token Measures Needed to Support the Alert Expression**

If you prefer to manually register the token measures, the `regTokenMeasure` must be run with `-FNHBI` option if in a global domain environment. This allows the token

measures to have different values across subdomains. The token measure requires a value to the measure while registering. In the following example, the token measure is registered in the Master Domain and are made to be equal to pos (Weekly Sales) since pos has the same base intersection (item/store/week) and data type (real) as the System Forecast for level 1.

Example:

```
C:\Domains\RDF>regTokenMeasure -d . -add sf01=pos -fnhbi
```

---

**Note:** Do not do this step if the batch has already been generated since the batch will have automatically registered sf01.

---

## Step 2

### Register the Alert Measure

The next step in the process is to register the alert measure in the Master Domain. In the following example, an alert with the name of rdf\_a1\_1 with label of Alert1level1 is being registered.

Example:

```
C:\Domains\RDF>regmeasure -d $DOMAIN_DEST_DIR -add "rdf_a1_1" -label
"Alert1level1" -baseint "itemstr_" -db "data/myalerts" -type boolean -navalue false
```

## Step 3

### Register the Expression for the Forecast Approval Alert

The alertmgr utility is used to register the alert and the alert expression. In the following example, the alert expression first filters out time series with low volume sales (items with forecasts less than three units). This alert compares the System Forecast in the first three weeks in the forecast horizon with last approved forecast for the same three weeks. If the values are within a 33% range, the full forecast horizon is set to automatic approval, otherwise the Alert is triggered. This is all done in batch, so the Alert Manager is NOT necessary to apply the alert. For intersections that do not qualify for automatic approval, the Approval Comment on the Approval Worksheet in the Forecast Approval workbook will contain *refused by alert*. You may use the Alert Manager to insert this alert into the workbook to display the intersections that have the alert flag set to **true**.

Example:

```
C:\Domains\RDF> alertmgr -d . -register "rdf_a1_1" -category "FORECAST_
APPROVAL" -categoryLabel "Alert1level1" -expression "rdf_a1_
1=if(tssum(@sf01,index([clnd].[week],flookup(lfsXLXB.level([data].[flvl]+[prod].top),[
data].[flvl].[flvl01])),
index([clnd].[week],flookup(lfsXLXB.level([data].[flvl]+[prod].top),[data].[flvl].[flvl01]
))+3)>=3.0,
abs(1-tssum(@sf01,index([clnd].[week],flookup(lfsXLXB.level([data].[flvl]+[prod].top),
[data].[flvl].[flvl01])),
index([clnd].[week],flookup(lfsXLXB.level([data].[flvl]+[prod].top),[data].[flvl].[flvl01]
))+3)/(tssum(lappf01XB,index([clnd].[week],flookup(lfsXLXB.level([data].[flvl]+[prod]
.top),[data].[flvl].[flvl01]))+3,index([clnd].[week],flookup(lfsXLXB.level([data].[flvl]+[p
rod].top),[data].[flvl].[flvl01])))+0.001))>.33,false)"
```

---

---

## Adding New Local Domains

This chapter provides an overview on adding new local domains to an existing RDF global domain. New local domains can be added using the RPAS `reconfigGlobalDomainPartitions` utility. It is important to keep in mind that as new local domains are added, they must be added such that the RDF partitioning requirements continue to be met. This means each new local domain can only contain one position along the partition dimension.

When new local domains are added, the following additional scripts must be executed, which are located in the `/bin` directory of `$RPAS_HOME`:

### **loadCurveParameters.ksh**

This script is used to load the Curve data parameter measures including Profile Data Source, Default Source Profile, Default Profile Approval Method, Training Window Method, and Normal Value. This action is typically performed within the plug-ins at domain creation time, however, when you add a new local domain to an existing domain environment, the plug-ins are not run, and therefore this script performs that action manually.

Usage:

```
loadCurveParameters -d fullPathToDomain -s fullPathToNewSubdomain
```

### **loadRDFParameters.ksh**

This script is used to load the RDF data parameter measures including Default Required Method, Default Source Level, Data Plan, Seasonal Profile and Spreading Profile. This action is typically performed within the plug-ins at domain creation time, however, when you add a new local domain to an existing domain environment, the plug-ins are not run, and therefore this script performs that action manually.

Usage:

```
loadRdfParameters -d fullPathToDomain -s fullPathToNewSubdomain
```



---

---

## 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.

Refer to "[Internationalization Considerations for RETL](#)" for information about internationalization and RETL.

### Translation

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

- Graphical user interface (GUI)
- Error messages

The following components are not translated:

- Documentation (online help, release notes, installation guide, user guide, operations guide)
- Batch programs and messages
- Log files
- Configuration tools
- Reports
- Demonstration data
- Training materials

The user interface has been translated into the following languages:

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

- Hungarian
- Italian
- Japanese
- Korean
- Polish
- Russian
- Spanish
- Swedish
- Turkish

---

---

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

---

---

---

---

## RPAS and RDF Integration with RMS

The strategy for the extraction of foundation data from Retail Merchandising System (RMS) is for the extract programs (RMSE) to provide flat files in a generic format. For each solution that will use this data, transformation scripts are used to reformat the data as needed to produce a file suitable for loading into the application. For the instances of data coming from RPAS to non-RPAS applications, extract programs are specific to the application in need of the data. Other scripting languages are then used (Perl or AWK) to perform additional data formatting.

This appendix summarizes the following:

- [Environment Variable Setup](#)
- [RDF Transformation Programs](#)
  - [Transformations of Merchandise Hierarchy Data](#)
  - [Transformations of Location Hierarchy Data](#)
  - [Transformations of Calendar Hierarchy Data](#)
  - [Transformations of Daily Sales and Issues Data](#)
  - [Transformations of Weekly Sales and Issues Data](#)
  - [Transformations of Store Open Date Data](#)
  - [Transformations of Store Close Date Data](#)
  - [Transformations of Out of Stock Indicator Data](#)
- [RDF Transformation Matrix](#)
- [Loading Transformed RMS Data into RDF](#)
- [Common Programs for Extracts](#)
- [Extract of Forecast Data for RMS](#)
- [Load of Extracted Forecast Data and Standard Deviations to RMS](#)
- [Extract of Diff Profile Data for Allocation](#)
- [Extract of Store Grade Data for RMS](#)
- [RDF Extract Matrix](#)

Specifics on the usage of RMS extract programs (RMSE's) within the RDF transformation programs are beyond the scope of this document. See the *Oracle Retail Merchandising System Operations Guide* for more information on the RMS extract programs.

---



---

**Note:** For integration compatibility information, see the *Oracle Retail Predictive Application Server Installation Guide*.

---



---

## Environment Variable Setup

In addition to any variables identified in the RMS integration documentation, the transformation and/or extract programs require the environment variables listed in the following table.

Environment Variable	Description
\$RPAS_INTEGRATION_HOME	Identifies the location of the integration scripts when /common/header.ksh is run. This variable is used for all integration scripts packaged with the ARPOPlatform except those included in RFX (see "\$SRDF_HOME").
\$TO_RPAS	The staging area for the data to be loaded into RPAS. This directory should be located at the same level as the root of the RPAS domain. For example, if the domain RDF is located in Domains directory.  (example: /Domains/RDF), then \$TO_RPAS should be located at the same level as RDF (example: /Domains/to_rpas).
\$FROM_RPAS	The staging area for the data extract out of RPAS. This directory should be located at the same level as the root of the RPAS domain. For example, if the domain RDF is located in Domains directory.  (example: /Domains/RDF), then \$FROM_RPAS should be located at the same level as RDF (example: /Domains/from_rpas).
\$RDF_HOME	Identifies the location of the root of the RFX directory. The RFX directory packaged with the ARPOPlatform should be added to the location RFX directory packaged with the RMS RETL programs.
\$RI_RMSVERSION	Identifies the version of RMS. If this variable is not set, the integration scripts assume an RMS version of 13. Set the value of this environment variable to 13.
RFX_HOME	The directory that holds the RETL executable files.
RETL_JAVA_HOME	The directory that contains Java. See the latest version of the <i>Oracle Retail Extract, Transform, and Load Programmer's Guide</i> for the version of Java that corresponds to the version of RETL used.
PATH	To run RETL, include these items in the PATH variable: \$RETL_JAVA_HOME/bin:\$RETL_JAVA_HOME/jre/bin:\$RFX_HOME/lib:\$RFX_HOME/bin:\$PATH
LIBPATH	To run RETL, include these items in the LIBPATH variable: \$RETL_JAVA_HOME/jre/bin:\$RETL_JAVA_HOME/jre/bin/classic:\$RFX_HOME/bin:\$LIBPATH
RMS_RPAS_HOME	This variable needs to be set in order to maintain compatibility with RMS. The required value is \$RDF_HOME.
RESOURCE_DIR	This directory contains resource information to support multi-language capability for calendar-related fields as well as the Unassigned value present in some schema.  The default (and recommended) value is \$RDF_HOME/resources.
PREFERRED_LANG	This value indicated the language that the calendar labels will be shown  The default is english (_en), but any value from \$RDF_HOME/resources/SupportedLanguages.txt where <i>column 4 = Yes</i> is supported, provided the corresponding retl_msgs.* file is present.

Environment Variable	Description
RDF_SCHEMA_DIR	This directory contains the RFX or RETL schemas. This variable is used when converting the Unassigned value in the schemas to a locale specific language. The default (and recommended) value is \$RDF_HOME/ <i>rfx/schema</i> .
UPGRADE_BACKUP_DIR	This directory is used to backup existing schema information prior to converting the Unassigned value in the schemas to a locale specific language. The default (and recommended) value is \$RDF_HOME/ <i>schema_backup</i> .

## RDF Transformation Programs

This section describes the RDF transformation scripts.

### Common Program for All Transformations

The `rdft.ksh` script runs all of the necessary data extraction and transformation scripts (`rmse_*.ksh` and `rdft_*.ksh`, respectively) that are needed to produce the files to be loaded into RPAS/RDF/Planning. Most of these scripts are run in parallel (as background jobs).

### Usage

```
rdft.ksh [-x] [-c] [-sd startDate] [-ed endDate] [-d dir]
```

### Arguments

The following table lists and describes the arguments for these scripts.

Argument	Description
-x	This option will cause the execution of the RMS data extraction wrapper ( <code>rmse.ksh</code> ) to be skipped.
-c	This option will cause <code>FILE_DATE</code> in <code>rmse_config.env</code> to be set to the current date instead of using <code>VDATE</code> .
-sd	This option sets the start date for optionally filtering out records based on date. Records with dates prior to this date will be excluded from loading into RDF. The date needs to be in the format <code>YYYYMMDD</code> .
ed	This option sets the end date for optionally filtering out records based on date. Records with dates after this date will be excluded from loading into RDF. The date needs to be in the format <code>YYYYMMDD</code> .
-d	This option will cause all programs executed by <code>rdft.ksh</code> to be obtained from the <i>dir</i> directory.

## Transformations of Merchandise Hierarchy Data

`rdft_merchhier.ksh` is the primary script used to build the data for RPAS from the RMS Merchandise Hierarchy tables. The schema used to produce the output file depends on the attributes and differentiator settings in RMS:

Case	Settings	Description
1	If <code>PROD_ATTRIBUTES_ACTIVE = false</code> and <code>DIFFS_ACTIVE = false</code>	Then <code>rdft_merchhier.base.schema</code> will be used to produce the file. In this case, attributes and diff fields will not be included in the merchandise hierarchy file.
2	If <code>PROD_ATTRIBUTES_ACTIVE = true</code> and <code>DIFFS_ACTIVE = false</code>	Then <code>rdft_merchhier.attributes.schema</code> will be used to produce the file. This schema must be manually edited to support a specific attribute model and must be kept in sync with <code>rmse_attributes.schema</code> and <code>rmse_attributes.ksh</code> (see the RMSE end user documentation).
3	If <code>PROD_ATTRIBUTES_ACTIVE = false</code> and <code>DIFFS_ACTIVE = true</code>	Then <code>rdft_merchhier.schema</code> will be used to produce the file. In this case, diff fields will be included in the merchandise hierarchy file.
4	If <code>PROD_ATTRIBUTES_ACTIVE = true</code> and <code>DIFFS_ACTIVE = true</code>	Then an error will result. In this release, the combination of diffs and attributes is not supported.

Intermediate schema and scripts which may be used (depending on configuration options) to produce the merchandise hierarchy file:

- `rdft_diff.domain.schema`
- `rdft_merchdiff.domain.schema`
- `rdft_merchhier_diff_trans.ksh`
- `rdft_merchhier_split_by_domain.ksh`
- `rdft_clean_partition.ksh`

Additional merchandise hierarchy support for issue domains is provided in `rdft_item_loc.ksh`. This script is designed to produce a full item list for issues domains, only containing items that exist in the warehouses.

---



---

**Note:** Issues-specific data transformation functionality is triggered based on the issues setting in RMS (`ISSUES_ACTIVE` must be set to **true**).

---



---

The script `rdft_clean_partition.ksh` was designed to mimic some of the functionality of local domains prior to local domains being supported by RPAS. This script is not relevant to versions of RDF 12.0 or later. It is retained for use by customers on earlier versions of RDF.

## Transformations of Location Hierarchy Data

`rdft_orghier.ksh` is the primary script used to build the location data file needed for RPAS from the RMS Organizational Hierarchy Table.

The following constants may be modified in the script based on location hierarchy data requirements:

Modifiable Constant	Description
COMPANY_NAME	The label for the company position to be populated in the file.
COMPANY_ID	The name for the company position to be populated in the file.
STORE_CLASS_CONCAT	When set to <b>true</b> , causes the STORE_CLASS to be concatenated on the left of the STORE_CLASS_DESCRIPTION field in the final Store data output file.
ADD_AT_SIGN_TO_WH_DESC	When set to <b>true</b> , will cause the WHSE_NAME field in the Warehouse output file to have an @ prefix.
LONG_WAREHOUSE_RECORDS	When set to <b>true</b> , the Warehouse output records will consist of 16 fields. If it is <b>false</b> , the records will contain only four fields, WH, WHSE_NAME, COMPANY, and CO_NAME.

Intermediate schemas which may be used (depending on configuration options) to produce the location hierarchy file:

- `rdft_issues.schema`
- `rdft_issues_long.schema`
- `rdft_orghier_store.schema`

---



---

**Note:** Issues-specific data transformation functionality is triggered based on the issues setting in RMS (ISSUES\_ACTIVE must be set to **true**).

---



---

## Transformations of Calendar Hierarchy Data

`rdft_calhier.ksh` transforms the Calendar Hierarchy data extracted from RMS for loading into RPAS.

Configuration inputs to the script include:

- **DATE\_PREF** - The path to the file that contains text indicating whether the format of the Date Description field will be mm/dd/yyyy or dd/mm/yyyy. See the *Oracle Retail Merchandising System Operations Guide* for date format options.
- **LAST\_DOW** - The path to the file that contains a day of week name or abbreviation indicating which day of the week is considered to be the end of the week for the fiscal calendar being used at this installation.

## Transformations of Daily Sales and Issues Data

`rdft_daily_sales.ksh` produces the daily sales and issues data files based on regular, promotion, clearance, and issues.

The following constant may be modified in the script based on data requirements:

- **DOM\_START\_COL** - Defines the starting column position of the Domain ID in the RETL output schema. This is needed by `rdft_merchhier_split_by_domain.ksh` to split the files by domain ID. If the `OUTPUT_SCHEMA` file is modified, the value of `DOM_START_COL` may also require modification from the default value.

Intermediate schemas which may be used (depending on configuration options) to produce the sales and/or issues data file:

- `rdft_daily_sales.schema`

---

---

**Note:** Issues-specific data transformation functionality is triggered based on the issues setting in RMS (`ISSUES_ACTIVE` must be set to **true**).

---

---

## Transformations of Weekly Sales and Issues Data

`rdft_weekly_sales.ksh` produces the weekly sales and issues data files based on regular, promotion, clearance and issues.

The following constant may be modified in the script based on data requirements

- **DOM\_START\_COL** - Defines the starting column position of the Domain ID in the RETL output schema. This is needed by `rdft_merchhier_split_by_domain.ksh` to split the files by domain ID. If the `OUTPUT_SCHEMA` file is modified, the value of `DOM_START_COL` may also require modification from the default value.

Intermediate schemas which may be used (depending on configuration options) to produce the sales and/or issues data files:

- `rdft_weekly_sales.schema`

---

---

**Note:** Issues-specific data transformation functionality is triggered based on the issues setting in RMS (`ISSUES_ACTIVE` must be set to **true**).

---

---

## Transformations of Store Open Date Data

`rdft_open_date.ksh` produces the Store/Warehouse Opening Date data file.

Intermediate schema used to produce the store open date data files:

- `rdft_open_date.schema`

## Transformations of Store Close Date Data

`rdft_close_date.ksh` produces the Store/Warehouse Closing Date data file.

Intermediate schema used to produce the store closing date data files:

- `rdft_close_date.schema`

## Transformations of Out of Stock Indicator Data

`rdft_outofstock.ksh` produces the Store and Warehouse (issues) Out of Stock Indicator data extracted from RMS.

Intermediate schema and scripts which may be used (depending on configuration options) to produce the Out of Stock Indicator data file:

- `rdft_outstock_split_by_domain.awk`
- `rdft_outofstock.schema`
- `rdft_outofstock_issues.schema`
- `rdft_outofstock_sales.schema`

## RDF Transformation Matrix

The following matrix identifies the transformation scripts and schemas used for each the hierarchy and data files produced for RDF:

Directory	Script or Schema Name	Merchandise Hierarchy	Location Hierarchy	Calendar	Daily Sales & Issues	Weekly Sales & Issues	Out of Stock Indicator	Store Open Dates	Store Close Dates
rxf/lib	rdft_merchhier_diff_trans.ksh	X							
	rdft_merchhier_split_by_domain.ksh	X							
	rdft_outofstock_split_by_domain.ksh						X		
rxf/schema	rdft_close_date.schema								X
	rdft_daily_sales.schema				X				
	rdft_diff.domain.schema	X							
	rdft_merchierdiff.domain.schema	X							
	rdft_merchier.attributes.schema	X							
	rdft_merchhier.base.schema	X							
	rdft_merchhier.domain.schema	X							
	rdft_merchhier.schema	X							
	rdft_open_date.schema							X	
	rdft_orghier_issues.schema	X							
	rdft_orghier_issues_long.schema	X							
	rdft_orghier_store.schema	X							
	rdft_outofstock.schema						X		
	rdft_outofstock_issues.schema						X		

Directory	Script or Schema Name	Merchandise Hierarchy	Location Hierarchy	Calendar	Daily Sales & Issues	Weekly Sales & Issues	Out of Stock Indicator	Store Open Dates	Store Close Dates
	rdft_outofstock_sales.schema						X		
	rdft_weekly_sales.schema					X			
rfx/src	rdft_ksh	X	X	X	X	X	X	X	X
	rdft_calhier.ksh			X					
	rdft_clean_partition.ksh	X							
	rdft_close_date.ksh								X
	rdft_daily_sales.ksh				X				
	rdft_item_loc.ksh	X							
	rdft_merchhier.ksh	X							
	rdft_open_date.ksh							X	
	rdft_orghier.ksh	X							
	rdft_outofstock.ksh						X		
	rdft_weekly_sales.ksh					X			

## Loading Transformed RMS Data into RDF

### renameRETLFiles.ksh

After the transformation of the RMS hierarchy and history data, the data files must be combined and renamed to match the appropriate measure names expected by RDF. This is accomplished by a script, `renameRETLFiles.ksh`. Usage of this script is optional, but recommended. The script reads the data files produced by the RETL transformation, combines and renames the data files, and writes the output to the domain's input folder. Existing data in the domain's input folder is backed up prior to the writing the new files.

### Usage

```
renameRETLFiles.ksh -dataDir <data dir> -domInput <domain's input dir>
```

### Arguments

The script has two required arguments:

Argument	Description
-dataDir	Required. The path to the data directory. This directory contains the RMS data transformed by the RDF RETL scripts. This is typically <code>\$RDF_HOME/data</code> .
domInput	Required. The input directory of the domain in which the transformed data is to be loaded. The directory is typically the input directory under the domain root.

### loadRETLMeasures.ksh

This script is another convenience script. Its usage is optional. It calls `loadHierarchy` and `loadMeasure` on the files produced by the RETL transformation and renamed by `renameRETLFiles.ksh` to load data into a specified RDF domain. It loads:

- The Calendar, Merchandise, and Location hierarchies
- Standard measures such as daily sales (dpos), regular sales (rsal), promo sales (psal), and clearance sales (csal).
- Optional measures. These can be anything, but are typically user-specified, such as out of stock indicators (outage), store opening dates (stropen), store closing dates (strclose), and so on. The list is not limited to measures associated with the RETL transformation – any valid measure may be specified.

### Usage

```
loadRETLMeasures -d <domain path> [-m measure1] [-m measure2 etc.]
```

### Arguments

The script accepts the following arguments:

Argument	Description
-d	Required. The path to the domain. This script (like all RPAS-based applications) assumes the data files are in the domain's input directory.
-m	Optional. The name of a measure to load. The script assumes the data file is in the domain's input directory

## Common Programs for Extracts

`config.ksh` is a configuration directory that requires both the RMS version being integrated and the backup action to be defined.

### Arguments

The following optional arguments are available:

Argument	Description
Name of the domain	Defaults to directory name
Number of the domain	Defaults to the 2 last digits of the directory name
Format of timestamp attached to logs and processed input files	Defaults to: (date + "%b%d%a%I%M%p") (example: Aug02Thu0111PM)
Data Drop	Defaults to ../../to_rpas
Data Export	Defaults to ../../from_rpas
Log Drop	Defaults to ./logs
Error Drop	Defaults to ./err
Reclass Data	Defaults to ../reclass_data

### functions.ksh

This script file contains ksh functions that are used by scripts in [DOM]/scripts. It should be sourced, not executed in order to preserve environment variables.

### header.ksh

This script should be run at the beginning of any implementation-specific script to setup function libraries, environment, and platform-specific routines.

## Extract of Forecast Data for RMS

`rdf_e_rms.ksh` extracts forecast demand value and standard deviation (cumulative interval) at both day and week aggregations from an RDF domain.

### Arguments

The script accepts the following arguments:

Argument	Description
-t	<Domain Type> (S for sales, I for issues)
-w	<Data Width> ([7...18], defaults to 14)
-d	<Domain> (defaults to current directory)
-n	<Domain Number> (defaults to last two digits of domain)
-s	<Start Date> (format YYYYMMDD, if not set, defaults to RPAS_TODAY; if RPAS_TODAY not set, then defaults to system date)

### Output Files

`${RPAS_EXPORT}/d<s|i>demand.<Domain Number>` (demand at day)

`${RPAS_EXPORT}/w<s|i>demand.<Domain Number>` (demand at week)

The following table provides information about the output file data format.

Field	Start	Width	Format
Day   EOW Day	1	8	Alpha
Product ID	9	25	Alpha
Location ID	34	20	Alpha
Demand	54	14	Alpha
Std. Dev. Demand	68*	14*	Numeric (floating point, 4 decimal digits with decimal)

\* Width of Demand and Std. Dev. Demand may be overridden with the `-w` parameter; stated values Demand width and Std. Dev. Demand start and width are based on default width of 14.

---

**Note:** The following must be defined in the shell environment prior to calling this script:

- RPAS\_HOME
  - RPAS\_INTEGRATION\_HOME
- 

## Editing for Simple Domains

If you have a simple domain, then the `functions.ksh` script needs to be edited. Under the `CreateWeek2DayArray ()` function call, edit this line by removing the `-fnhbi` flag.

Edit from:

```
regmeasure -d $RPAS_DOMAIN -add WEEK2DATE -type date -baseint
week -fnhbi -db data/hmaint
```

to:

```
regmeasure -d $RPAS_DOMAIN -add WEEK2DATE -type date -baseint
week -db data/hmaint
```

## Load of Extracted Forecast Data and Standard Deviations to RMS

`rmsl_forecast.ksh` pulls the daily/weekly forecast items into RMS.

During the loading of each domain file the following steps are performed:

1. Truncate the partition in the RMS forecast table which corresponds to the domain ID.

---

**Note:** Partition names should always be in the format:  
[tablename]\_[domainID]

---

2. Append a domain field and insert the `domain_id` into each record.
3. Load the forecast data into the RMS forecast table.

**Example:** `rmsl_rpas_forecast.ksh daily | weekly`

Intermediate schemas which may be used (depending on configuration options) to produce the forecast data files:

- rmsl\_forecast\_daily.schema
- rmsl\_forecast\_weekly.schema

## Extract of Diff Profile Data for Allocation

`profile_e_alloc.ksh` extracts Curve diff profiles for use by Allocation.

### Arguments

The script accepts the following arguments:

Argument	Description
-p	<Profile Number>
-m	<Mask Measure> (Optional mask; only positions for which the mask value is non-NA will be exported.)
-w	<Data Width> ([7...18], defaults to 12)
-d	<Domain> (defaults to current directory)
-n	<Domain Number> (defaults to last two digits of domain)

### Output file

`${RPAS_EXPORT}/d1<Product Level>.<Domain Number>`

---

**Note:** Where Product Level is the Aggregation intersection's Prod dimension.

---

The following table provides information about the output file data format.

Field	Start	Width	Format
Product ID	1	25	Alpha
Location ID	26	20	Alpha
Diff ID (optional)	46	36	Alpha
Quantity	82	12*	Numeric (floating point, 4 decimal digits, no decimal)*
Std. Dev. Demand	68*	12*	Numeric (floating point, 4 decimal digits with decimal)

\* Quantity width may be overridden with the -w parameter.

---

**Note:** The following must be defined in the shell environment prior to calling this script:

- RPAS\_HOME
  - RPAS\_INTEGRATION\_HOME
-

## Extract of Store Grade Data for RMS

`grade_e_rms.ksh` extracts store grades for use by RMS.

### Arguments

The script accepts the following arguments:

Argument	Description
-t	<Timestamp> (YYMMDDTTTT). This value corresponds to the timestamp of the Cluster Membership measure (clpm+<Timestamp>) to be extracted
-d	<Domain> (defaults to current directory)
-n	<Domain Number> (defaults to last two digits of domain)

### Output File

`${RPAS_EXPORT}/gr<Timestamp>.<Domain Number>`

---

**Note:** The following must be defined in the shell environment prior to calling this script:

- `RPAS_HOME`
  - `RPAS_INTEGRATION_HOME`
- 

The following tables list the output file data formats:

Header Records
FHEAD
Line ID Number
GRADU

Detail Records	Description
FDETL	Record Identifier
Line Sequence Identifier	NA
Grade Group ID Number	This value corresponds to the first 8 characters of the Cluster Run Name measure (clnam+<user-defined name>) set by the user in the Generate Cluster wizard in Grade. For integration with RMS, the Cluster Run Name must be populated with only numeric characters.
Grade Group	This value corresponds to the first N characters of the Cluster Run Name measure (clnam+<user-defined name>) set by the user in the Generate Cluster wizard in Grade. N is 20 for RMS version 11.0, and N is 120 for RMS version 12. The script determines the RMS version from the environment variable <code>RI_RMSVERSION</code> . RMS version 12 is assigned by default.
Store ID	NA
Grade Member Name	NA
FTAIL	Record Identifier)

Footer Records
Line ID Number
FDETL Line Total Number

## RDF Extract Matrix

The following matrix identifies the extract scripts and schemas used for each the data files produced for RMS.

Directory	Script or Schema Name	Forecasts and Standard Deviations	Diff Profiles
common	config.ksh		
	functions.ksh	X	
	header.ksh	X	X
curve	profile_e_alloc.ksh		X
grade	grade_e_rms.ksh		
plan	Plan_e_alloc.ksh		
	Plan_e_price.ksh		
	Plan_e_plcblwdm.ksh		
	Plan_e_poblwdm.ksh		
rdf	rdf_e_rms.ksh	X	
	rmsl_forecast.ksh	X	
	rmsl_forecast_daily.schema	X	
	rmsl_forecast_weekly.schema	X	

## Internationalization Considerations for RETL

These sections describe internationalization considerations for RETL:

- [Calendar Data](#)
- [The Unassigned Value in Schemas](#)

### Calendar Data

Calendar data (that is names of weeks and months) is translatable into Oracle supported languages. Enabling this feature requires setup of these two environment variables:

- export RESOURCE\_DIR=\$RDF\_HOME/resources
- export PREFERRED\_LANG=\_en

Note that for PREFERRED\_LANG, the default is english (\_en), but any value from \$RDF\_HOME/resources/SupportedLanguages.txt where *column 4 = Yes* is supported, provided the corresponding retl\_msgs.\* file is present.

Once these variables are setup, then the translation takes place for RETL. No separate script is required.

## The Unassigned Value in Schemas

The `Unassigned` value in schemas can be converted to a locale specific equivalent. To enable this feature, set these environment variables:

- `export RESOURCE_DIR=$RDF_HOME/resources`
- `export PREFERRED_LANG=_en`
- `export RDF_SCHEMA_DIR=$RDF_HOME/rfx/schema`
- `export UPGRADE_BACKUP_DIR=$RDF_HOME/schema_backup`

Note that for `PREFERRED_LANG`, the default is english (`_en`), but any value from `$RDF_HOME/resources/SupportedLanguages.txt` where *column 4 = Yes* is supported, provided the corresponding `retl_msgs.*` file is present.

After the environment variables are set, then execute this script: `$RDF_HOME/rfx/etc/update_schema_nullvalues.ksh`.

This script backs up existing schemas into `UPGRADE_BACKUP_DIR` and then replaces the `Unassigned` value with a locale specific equivalent. This update preserves field widths.