Oracle® Retail Demand Forecasting Cloud Service

Implementation Guide

Release 19.0

F24923-16

October 2023

ORACLE®

Oracle Retail Demand Forecasting Implementation Guide, 19.0 for Windows

F24923-16

Primary Author: Melissa Artley

Contributing Author: Lin He

Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

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

Contents

3 RDF Configuration

4 RDF Cloud Service Extensibility

A Appendix: Rules to Populate Out-of-stock and Outlier Indicator Measures

B Appendix: RDF Cloud Service integration with RMF Cloud Service

C Appendix: RDF Cloud Service Integration with PDS and BDI

D Appendix: New Item Configuration Guide

E Appendix: RDF Cloud Service Integration with ORASE / RI

F Appendix: Configuring Batch Run

List of Figures

List of Tables

Send Us Your Comments

Oracle Retail Demand Forecasting Cloud Service Implementation Guide, Release 19.0.

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 Help Center web site. It contains the most current Documentation Library plus all documents revised or released recently.

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

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

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

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

Preface

This document provides critical information about the processing and operating details of Oracle Retail Demand Forecasting Cloud Service.

Audience

This document is for:

- Systems administration and operations personnel
- Systems analysts
- Integrators and implementers
- Business analysts who need information about Oracle Retail Demand Forecasting Cloud Service processes and interfaces

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle Retail Demand Forecasting Cloud Service 19.0 documentation set:

- *Oracle Retail Demand Forecasting Cloud Service Administration Guide*
- *Oracle Retail Demand Forecasting Cloud Service Release Notes*
- *Oracle Retail Demand Forecasting Cloud Service Starter Kit Guide*
- *Oracle Retail Demand Forecasting Cloud Service User Guide*
- Oracle Retail Predictive Application Server Cloud Edition documentation

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

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

Improved Process for Oracle Retail Documentation Corrections

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

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

https://docs.oracle.com/en/industries/retail/index.html

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

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

Oracle Retail Documentation on the Oracle Help Center (docs.oracle.com)

Oracle Retail product documentation is available on the following web site:

https://docs.oracle.com/en/industries/retail/index.html

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

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.

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

1

Introduction

Oracle Retail Demand Forecasting Cloud Service (RDF CS) provides accurate forecasts that enable retailers to coordinate demand-driven outcomes that deliver connected customer interactions. With a single view of demand, RDF CS provides pervasive value across retail processes, including driving optimal strategies in planning, increasing inventory productivity in supply chains, decreasing operational costs and driving customer satisfaction from engagement to sale to fulfillment. RDF CS, is a comprehensive solution that maximizes the forecast accuracy for the entire product lifecycle; with tailored approaches for short and long lifecycle products; the ability to adapt to recent trends, seasonality, out-of-stocks and promotions; and reflect the unique demand drivers of each retailer.

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.

Forecasting Challenges and RDF CS Solutions

A number of challenges affect the ability of forecast demand accurately including:

- Selecting the Best Forecasting Method
- Overcoming Data Sparsity through Escalation and Pooling Levels
- Forecasting Demand for New Products and Locations
- Managing Forecasting Results Through Automated Exception Reporting
- Incorporating the Effects of Promotions and Other Event-Based Challenges on Demand
- Support for Short Lifecycle Merchandise
- 53 Week Calendar

Selecting the Best Forecasting Method

One challenge to accurate forecasting is the selection of the best model to account for level, trending, seasonal, and spiky demand. Oracle Retail's automatic evaluation of several methods eliminates this complexity. The automated approach can pick the best fit method among a large selection, like Simple Exponential Smoothing, Holt Exponential Smoothing, Additive and Multiplicative Winters Exponential Smoothing, Croston's Intermittent Demand Model, and Seasonal Regression forecasting.

Another approach is to combine the output of the competing methods to create a more robust forecast and minimize the risk of overfitting

Overcoming Data Sparsity through Escalation and Pooling Levels

Demand at low levels, such as item/store is usually too noisy to identify clear selling patterns, both for baseline and promotional sales. In such cases, generating a reliable forecast requires analyzing historical data at a higher level (escalation or pooling levels) in the hierarchy in which demand patterns can be consistently detected. The forecasting components estimated at these high levels, like seasonality curves and promotion effects, are combined with low level information, like base demand and trend, to create the low level forecast that is needed to drive the supply chain.

Forecasting Demand for New Products and Locations

RDF CS also forecasts demand for new products and locations for which no sales history exists. There are several options for new products. First, there is the option to go on auto mode, and the user does not have to do anything. Another option is model the new products demand based on that of an existing similar product for which you do have a history. Forecasts for the new products are copied from one item or can be a combination of multiple items.

Managing Forecasting Results Through Automated Exception Reporting

The RDF CS end user is typically responsible for managing the forecast results for thousands of items, at hundreds of stores, across many weeks at a time. The Oracle Retail Predictive Application Server Cloud Edition (RPAS CE) platform provides users with an automated exception reporting process that indicates to you where a forecast value may lie above or below an established threshold, thereby reducing the level of interaction needed from you. The framework for exception management is implemented using multiple features.

First there is the exception dashboard profile, where the user can filter down to desired merchandise/locations to view a hit count and the variance from the desired value of the forecast. Based on that information, the user can launch in a workspace where she can review only the exceptions inside the product and locations space defined in the dashboard filter settings.

Once in the workspace, the user navigates to flagged positions using the workbook alerts which are synchronized with the exception dashboards. When an exception is resolved, the result is committed to the domain, and the dashboard exception count - upon refresh - reflects the change.

Incorporating the Effects of Promotions and Other Event-Based Challenges on Demand

Promotions, non-regular holidays, and other causal events create another significant challenge to accurate forecasting. Promotions such as advertised sales and free gifts

with purchase might have a significant impact on a product's sales history, as can fluctuating holidays such as Easter.

The causal forecasting functionality estimates the effects that such events have on demand. The results are used to predict future sales when conditions in the selling environment are similar. This type of advanced forecasting identifies the behavioral relationship of the variable you want to forecast (sales) to both its own past and explanatory variables such as promotion and advertising.

Suppose that your company has a large promotional event during the Back To School season each year. The exact date of Back To School varies from year to year, as a result, the standard time-series forecasting model often has difficulty representing this effect in the seasonal profile. The Promotional Forecasting module allows you to identify the Back To School season in all years of your sales history, and then define the upcoming Back To School date. By doing so, you can causally forecast the Back To School-related demand pattern shift.

Demand Transference

What is Demand Transference?

When new items are added to a retailer's assortment, the existing items' demand may be negatively impacted. Conversely, if items are removed from an assortment, the remaining items may experience a boost in demand, because there is less competition. We call this demand transference due to assortment changes.

Demand transference across items in an assortment is a challenge for all retailers. How one item effects the performance of another is difficult to predict, and can significantly impact an assortments performance overall.

Enabling demand transference in RDF CS adds to the already extensive capabilities to create more accurate and robust forecasts by incorporating the impact of assortment changes. Accurate forecasts translate to high service levels and fewer lost sales, which in turn means better margin and improved customer satisfaction. With this release, RDF CS is enhanced to layer the demand transference impact on top of the base demand, trend, seasonality, as well as price and promo information, to create a holistic version of the forecast.

Support for Short Lifecycle Merchandise

Short lifecycle items have the unique trait that they sell for a relatively short period of time and then never again. This type of merchandise can be divided as fashion items, and items that have replacements. For fashion items, the demand is modeled based on items that started selling around the same time of year in the past years. For instance, a spring collection for this coming year, is modeled based on a Spring collection that started selling in February in the past year.

The items that replace other items are treated differently. The demand for an item that will start selling is going to be modeled after the demand of the item that it is replacing.

53 Week Calendar

For the majority of retailers, the business is managed using a calendar (364 days organized into 13 week quarters) that periodically includes an extra 53rd week so that the year end stays in about the same time of the year. It is useful to have some control over how this 53rd week will be managed within the forecasting system's time

dimension. Management of this issue causes customers the pain, time and cost of configuring their data every few years that this happens.

The problem described has two implications. The first case is when two years - each with 52 weeks - of historical sales are available, and the retailer needs to forecast for the following year, which has 53 weeks. The second case is when one of the years of historical sales has 52 weeks, and the other has 53 weeks.

The correction for the extra week happens as part of forecasting.

RDF CS creates a measure, Week 53 indicator, with the base intersection that is entered by the user in the RDF plug-in. This measure has to be loaded by the user.

There is also the measure, Week 53 Options, that indicates how to calculate the forecast value for a week that was flagged as 53rd or extra week. This measure is present in the Forecast Setup View.

Forecasting Process

The forecasting process represents a next generation approach engineered to provide transparency, responsiveness and accuracy through the application of retail sciences using the scale of our modern Retail Cloud Platform.

- ***Transparency*** enables analytical processes and end-users to understand and engage with the forecast. This is accomplished by representing the demand model as the decomposition of intuitive components that include base rate of demand, seasonality and causal effects. The forecasting process provides transparency to the final results, individual model components and underlying decisions by the system and end-user.
- ***Responsiveness*** enables the coordination and simulation of demand-driven outcomes using forecasts that adapt immediately to new information and without a dependency on batch processes. This is accomplished by separating the calculation of the forecast from the analytical processes that determine components within the forecasting model.
- ***Accuracy*** enables retailers to deliver connected customer interactions while driving efficiencies to increase profits. Maximizing forecast accuracy is paramount to RDF CS. This is accomplished through the application of best-fit sciences throughout the forecasting process.

Process Summary

Following is a summary of the forecasting process:

1. Prepare Reference Data

   The purpose of this step is to prepare reference data for subsequent estimation, pruning and escalation processes. The emphasis in the preparation processes is to treat anomalies in historical data; such as out-of-stock, outliers and promotions; where the objective is to increase reliability of the reference data. For long-lifecycle items where data tends to be reliable over long time periods, the anomalies are corrected. For short-lifecycle items where data tends to be unreliable over short periods, the anomalies are omitted.

2. Estimated Demand Parameters

   The purpose of this step is to estimate all demand parameters and at all possible escalation levels. An escalation level represents a grouping of items and locations for robust parameter estimation to overcome sparsity and sensitivity. Escalation

levels can be tied to explicit hierarchy levels (for example; subclass/region) or flexible item/location groupings (for example; optimized analytical clusters). As each demand parameter is estimated, multiple machine learning methods are applied, individually optimized and evaluated for accuracy. The final model can represent the best-fit method or a robust method calculated as an intelligent blending of multiple methods weighted by accuracy.

3. Prune

   The purpose of this step is to prune escalation levels that do not pass analytical quality checks. These include data, estimation and correlation quality checks. The result is a candidate pool of high quality parameter estimates for the escalation process.

4. Escalate

   The purpose of this step is to select the demand parameter estimate for each component of the forecast model using the candidate pool of escalation levels. The escalation process reflects the optimal balance of richness and reliability.

5. Forecast

   The purpose of this step is to calculate the forecast through the application of demand parameter estimates from the analytical processes in conjunction with the known demand drivers and user-overrides. The demand model is completely responsive to changes in demand drivers and updates to the demand model itself (for example; user-defined override). This step also includes support for responsive new-item forecasting, with tailored approaches for new-item scenarios, such as dynamic, repeatable and similar assortments.

User Experience and Workflow

The user experience is delivered on our experience inspired RPAS Cloud Edition (RPAS CE) user interface (UI). RPAS CE provides end-users with a next generation cloud-native UI that purpose-built to accelerate intent into action for planners and forecasters. This includes interactive and visual dashboards to assess priorities, responsive and flexible workspaces to implement decisions and a coordinated exceptions framework that ties business process all the way from dashboard to cell.

The business process is engineered to maximize the productivity of end-users through exception-driven processes and emphasis on workflow simplification. The day-in-the-life processes begin with dashboard views that enable the end-user to assess the effectiveness and quality of their forecasts and prioritize exceptions. From the dashboards, the end-user is able to contextually launch into the appropriate workspace. For exception-driven processes, the end-user is guided to the point-of-resolution, with visibility to progress and the ability to iteratively work through forecasting priorities throughout the day.

Dashboard Views and Workspaces

The dashboard views and workspaces that support day-in-the-life forecasting workflows are summarized as follows:

- Forecast Overview Dashboard

  This dashboard leads with KPIs that provide macro-level insight into forecasting priorities and the effectiveness of the forecasts in driving demand-driven outcomes. This enables end-users to assess forecasting complexity drivers, such as

frequent promotions, and forecasting performance towards business objectives, such as fill rates.

- Forecast Scorecard Dashboard

  This dashboard provides insight to forecast accuracy (for example; MAPE, Bias) along with clear visibility to system performance and the impact of end-user contributions to the forecasting process. This enables forecast analysts and managers to identify forecast process improvement priorities.

- Exception Dashboards

  The exception dashboards represent the primary starting point for day-in-the-life processes. The short and long lifecycle forecasting processes each have a dedicated dashboard that enables end-users to efficiently drive decisions through focused exception-driven processes. From here, end-users are able to define the scope of exceptions to be managed through dashboard filters and launch directly to workspace views tailored for resolution. As exceptions are resolved, the dashboard is updated to enable end-users to iteratively work through forecasting priorities.

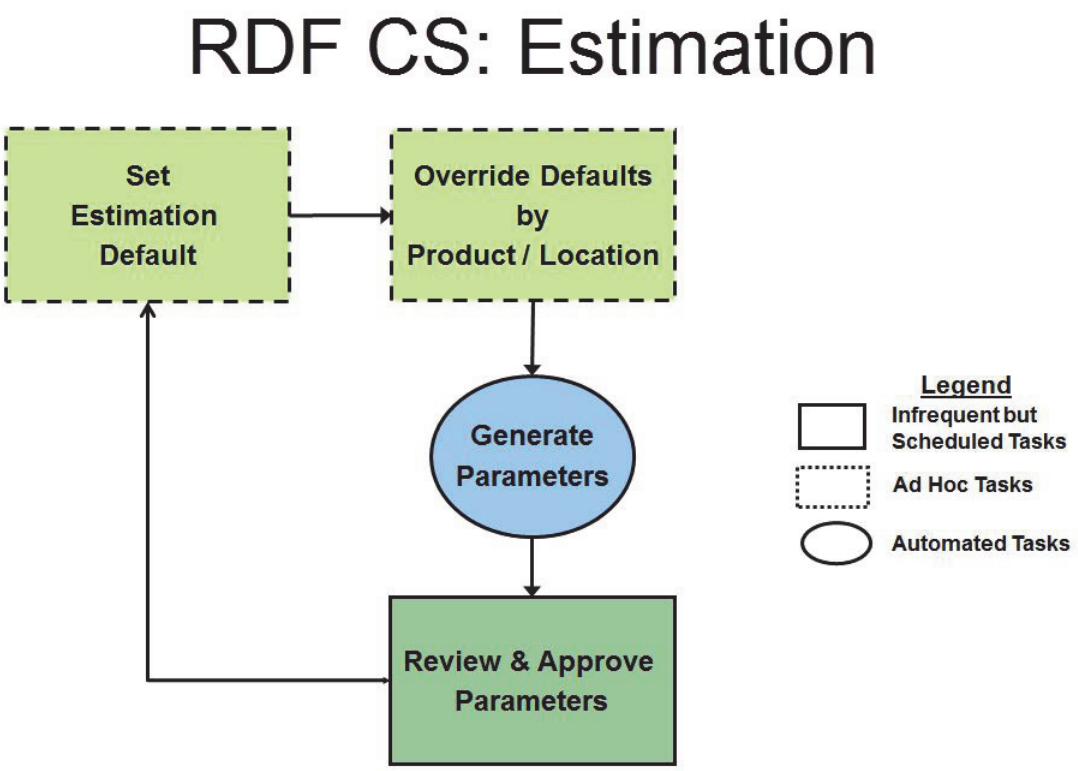
- Forecast Review Workspaces

  The forecast approval workspaces represent the primary point of interaction with the demand forecasts. The short and long lifecycle forecasting processes each have a dedicated forecast approval workspace to enable end-users to efficiently review, adjust, and approve their forecasts. This is supported by a rich set of decision support metrics and the ability to responsively simulate forecast updates based on new demand drivers and different forecasting methods. The workspace also features real-time alerts and dedicate exception management views that navigate end-users to the point resolution.

Forecast Engine

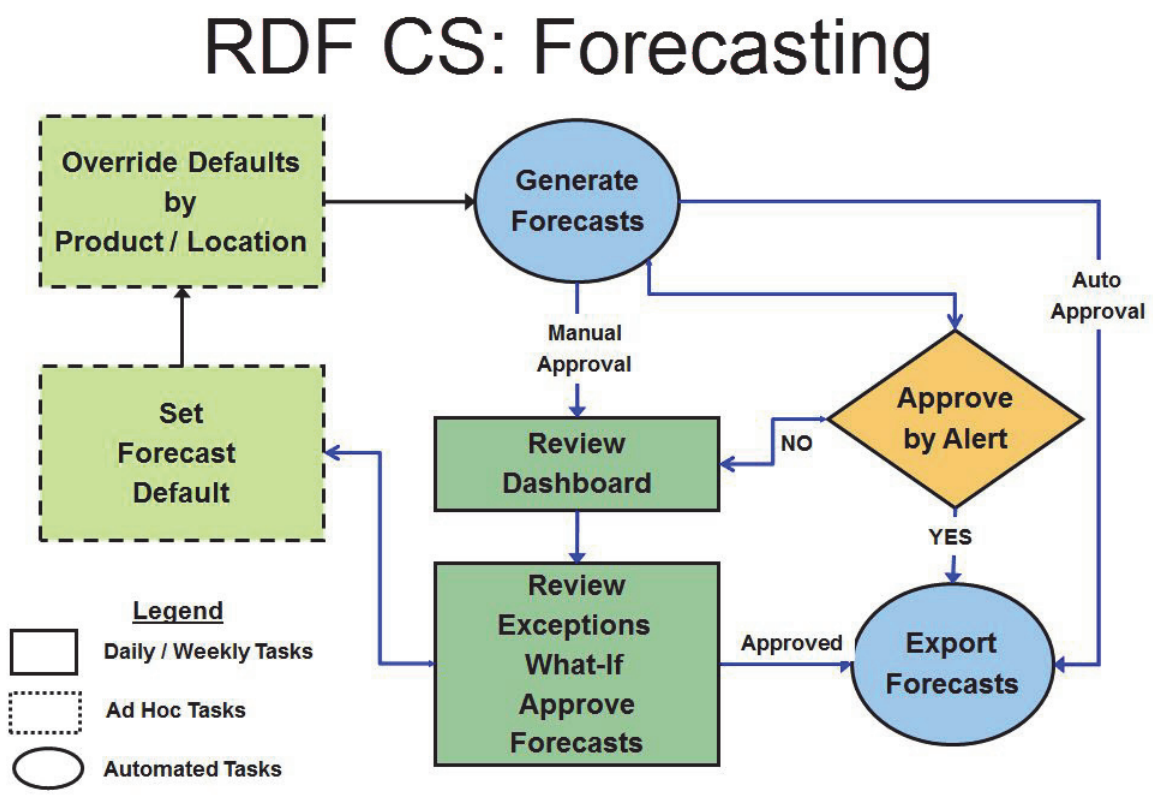
Not visible to the end user is the forecast engine, and all the tasks happening behind the scenes. The batch is split between estimation and forecasting. Estimation consists of the heavy data mining of historical demand to generate the necessary forecast parameters like seasonality, price and promo effects. Following are tasks comprised in the estimation workflow.

Figure 1–1 RDF CS Estimation



After estimation is run, the forecast parameters are computed, and everything is available to generate the forecast. Since most metrics are already pre-calculated during estimation, this step is very quick, allowing for extensive what-if in the Forecast review workbook. Below are the tasks related to generating, reviewing, and approving forecasts.

Figure 1–2 RDF CS Forecasting



2

Implementation Considerations

The following information must be considered before configuring Demand Forecasting Cloud Service:

- Configuration Considerations
- RDF Cloud Service Hierarchies
- RDF Cloud Service Input Data
- Integration
- User Roles and Securities
- Internationalization

Configuration Considerations

Before implementing RDF Cloud Service, an implementer should first answer the following questions:

1) Is my forecasted item Long Lifecycle (LLC) or Short Lifecycle (SLC)?

2) Are there any promotions that impact my forecast? If yes, how can I define the promotions?

3) What is the purpose of my forecast? To drive replenishment, allocation, or others?

4) Based on the purpose of my forecasting, which level should the forecast be generated on (sku/stor/week)? How many escalation levels are needed for the forecasting? Which level should the forecast be exported to?

5) What data is available to use for forecasting: rsal, psal, csal, Promotions, or Price?

6) What kind of preprocessing is needed: Outage, Outlier, Depromote, or Deseasonalize Smooth? Configuration details can be found in Preprocessing Configuration Process.

7) How do I want to handle New Items? Is there any product attribute information?

8) Do I want to integrate RDF Cloud Service with other Applications?

9) How do I want to partition the RDF Cloud Service domain?

10) If I want to use grouping in my escalation levels or pooling levels, how do I group my item/stores?

11) Do I have a foundation system to provide foundation (hierarchy) data?

12) Do I need to generate daily forecast, and/or both weekly and daily forecasts?

13) Do I have a foundation system to provide foundation (hierarchy) data?

Depending on the answers to the previous questions, the implementer can use the RDF Cloud Service plug-in to generate RDF Cloud Service configurations. For details about how to generate RDF Cloud Service configuration, refer to Chapter 3, "RDF Configuration". The generated RDF Cloud Service configuration can be customized to satisfy client specific requirement. For details about how to customize RDF Cloud Service configuration, refer to Chapter 4, "RDF Cloud Service Extensibility".

RDF Cloud Service Hierarchies

There are four type of hierarchies in RDF Cloud Service:

- Standard RPAS Hierarchy Files
- Static Internal RDF Cloud Service Hierarchy Loading Files
- Dynamic Internal RDF Cloud Service Hierarchy Loading Files
- RHS Hierarchies

Standard RPAS Hierarchy Files

Standard RPAS hierarchy, user provide the hierarchy loading files.

This is the foundation data to build any RPAS solution. Demand Forecasting Cloud Service requires the standard three hierarchy files, Calendar, Product, and Location. Also, additional sets of hierarchy files specific to different solutions are needed.

For information on the hierarchy files, see the following sections:

> **Note:** All of the following hierarchy files need to be provided. If the Group and Season Code hierarchy files are not available at the time of implementation, either the GA file or dummy positions need to be provided.

- Calendar Hierarchy File
- Product Hierarchy File
- Location Hierarchy File
- Group Hierarchy File
- Season Code Hierarchy File
- Products Attributes Hierarchy File

Calendar Hierarchy File

File name: clnd.csv.dat

File format: comma-separated values file

The following table describes the fields in this file.

Name	Label	Hierarchy Type	Parent
DAY	Day	Main	None
WEEK	Week	Main	DAY
MNTH	Month	Main	WEEK

Name	Label	Hierarchy Type	Parent
QRTR	Quarter	Main	MNTH
HALF	Half	Main	QRTR
YEAR	Year	Main	HALF
DOW	DAY OF WEEK	Alternate	DAY
WOYR	Week of Year	Alternate	WEEK
STDB	STD/BTA	UDA	WEEK

Example:

```
20170101,1/1/2017,W01_2017,1/6/2017,JAN_2017,"January, FY 2017",Q1_2017,"Quarter
1, FY 2017",S1_2017,"Season 1, FY 2017",A2017,FY2017,SUN,Sunday,WY01,Week 01
20170102,1/2/2017,W01_2017,1/6/2017,JAN_2017,"January, FY 2017",Q1_2017,"Quarter
1, FY 2017",S1_2017,"Season 1, FY 2017",A2017,FY2017,MON,Monday,WY01,Week 01
20170103,1/3/2017,W01_2017,1/6/2017,JAN_2017,"January, FY 2017",Q1_2017,"Quarter
1, FY 2017",S1_2017,"Season 1, FY 2017",A2017,FY2017,TUE,Tuesday,WY01,Week 01
20170104,1/4/2017,W01_2017,1/6/2017,JAN_2017,"January, FY 2017",Q1_2017,"Quarter
1, FY 2017",S1_2017,"Season 1, FY 2017",A2017,FY2017,WED,Wednesday,WY01,Week 01
20170105,1/5/2017,W01_2017,1/6/2017,JAN_2017,"January, FY 2017",Q1_2017,"Quarter
1, FY 2017",S1_2017,"Season 1, FY 2017",A2017,FY2017,THR,Thursday,WY01,Week 01
20170106,1/6/2017,W01_2017,1/6/2017,JAN_2017,"January, FY 2017",Q1_2017,"Quarter
1, FY 2017",S1_2017,"Season 1, FY 2017",A2017,FY2017,FRI,Friday,WY01,Week 01
20170107,1/7/2017,W02_2017,1/13/2017,JAN_2017,"January, FY 2017",Q1_2017,"Quarter
1, FY 2017",S1_2017,"Season 1, FY 2017",A2017,FY2017,SAT,Saturday,WY02,Week 02
```

Product Hierarchy File

File name: prod.csv.dat

File format: comma-separated values file

The following table describes the fields in this file.

Name	Label	Hierarchy Type	Parent
SKU	Item	Main	None
SKUP	Style/Color	Main	SKU
SKUG	Style	Main	SKUP
SCLS	Sub-Category	Main	SKUG
CLSS	Category	Main	SCLS
DEPT	Department	Main	CLSS
PGRP	Group	Main	DEPT
DVSN	Division	Main	PGRP
CMPP	Company	Main	DVSN
VNDR	Vendor	ALT	SKU
PAT1	Prod Attribute 1	UDA	SKU
PAT2	Prod Attribute 2	UDA	SKU
STA1	Style UDA 1	UDA	SKUG

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,Long Life Cycle Items,1,All Product,1000,Supplier 1
10000011,10000011Leather Loafer - Black 6.5 B,10000011,10000011Leather Loafer -
Black 6.5 B,10000009,10000009Leather
Loafer,122,122Loafer,1312,1312Casual*,1310,1310Footwear Women's*,1300,Group
1,1,Long Life Cycle Items,1,All Product,1000,Supplier 1
10000012,10000012Leather Loafer - Black 7 B,10000012,10000012Leather Loafer -
Black 7 B,10000009,10000009Leather
Loafer,122,122Loafer,1312,1312Casual*,1310,1310Footwear Women's*,1300,Group
1,1,Long Life Cycle Items,1,All Product,1000,Supplier 1
10000013,10000013Leather Loafer - Black 7.5 B,10000013,10000013Leather Loafer -
Black 7.5 B,10000009,10000009Leather
Loafer,122,122Loafer,1312,1312Casual*,1310,1310Footwear Women's*,1300,Group
1,1,Long Life Cycle Items,1,All Product,1000,Supplier 1
```

Location Hierarchy File

File name: loc.csv.dat

File format: comma-separated values file

The following table describes the fields in this file.

Name	Label	Hierarchy Type	Parent
STOR	Location	Main	None
DSTR	District	Main	STOR
REGN	Region	Main	DSTR
CHNL	Channel	Main	REGN
CHAN	Chain	Main	CHNL
COMP	Company	Main	CHAN
SFMT	Store Format	Alternate	STOR
STCL	Store Class	Alternate	STOR
SAT1	Store Attribute 1	UDA	SAT1
SAT2	Store Attribute 2	UDA	SAT2

Example:

```
1000,New York City,1000,US,1000,North America,1000,The Americas,1000,Bricks &
Mortar,100,JCB Trading Company,4,4,A,A
1010,Boston,1000,US,1000,North America,1000,The Americas,1000,Bricks &
Mortar,100,JCB Trading Company,5,5,A,A
1020,San Francisco,1000,US,1000,North America,1000,The Americas,1000,Bricks &
Mortar,100,JCB Trading Company,5,5,A,A
1030,Seattle,1000,US,1000,North America,1000,The Americas,1000,Bricks &
Mortar,100,JCB Trading Company,4,4,A,A
```

Group Hierarchy File

The group hierarchy is an internal application-specific hierarchy to divide item/stores into different grouping to use during parameter estimation and forecasting.You can customize this hierarchy during implementation and use the GA dataset hierarchy as a

reference. Users can add or change how many groups are allowed in the domain through modifying the group hierarchy data file.

File name: grph.csv.dat

File format: comma-separated values file

The following table describes the fields in this file.

Name	Description
GRPD	This is the grouping to use during estimation and forecast.

Example:

```
111,Time Series Group 111
112,Time Series Group 112
113,Time Series Group 113
114,Time Series Group 114
115,Time Series Group 115
```

Season Code Hierarchy File

Short Lifecycle items that start selling around the same time and have a similar seasonality curve can be grouped together and assigned a Season code. Each season code represents one or several weeks within a range of seasonal length. Refer to the section, "Season Code Setup" in the *Oracle Retail Demand Forecasting Cloud Service User Guide*.

You can customize this hierarchy during implementation and use the GA dataset hierarchy as a reference.

Users can change how many season codes are allowed in the domain by modifying the season code hierarchy data file. The definition of each season code can be done through four measures:

Name	Description
seabegin_SF_	seabgein_SF_ defines the beginning of the sales start range. Its value should be a position name of the woyr dimension
seaend_SF_	seaend_SF_ defines the end of the sales start range.Its value should be a position name of the woyr dimension.
sealenmin_SF_	sealenmin_SF_ defines the minimum of the season length. Its value should be an integer.
sealenmax_SF	sealenmax_SF_ defines the maximum of the season length. Its value should be an integer.

File name: seac.csv.dat

File format: comma-separated values file

The following table describes the fields in this file.

Name	Description
code	This is the season code grouping to use during SLC forecasting.

Example:

```
001,Season code 001
```

```
002,Season code 002
003,Season code 003
004,Season code 004
```

Products Attributes Hierarchy File

The product attributes hierarchy represents attributes associated with products. These attributes are used to group products within categories. This grouping is what consumer decision trees are built on and are used when showing dynamic rollups at item level.

This hierarchy is intended to capture all product attributes for all product types. The attributes are then assigned to individual products. This assignment is used when processing the dynamic rollups.

This hierarchy is intended to be customized for the individual retailer's needs.

Name	Label	Hierarchy Type	Aggs
PATV	Prod Attribute Value	Main	None
PATT	Prod Attribute	Main	PATV

File name: patr.csv.dat

File format: comma-separated values file

The following table describes the fields in this file.

Field	Description
Prod Attribute Value	The various values that an attribute might have. For example, the package type attribute might take the values bag, box, or convenience.
Prod Attribute	The name of a product attribute, such as brand, family type, flavor, grain, package type, size, or temperature.

Example:

```
patv,patv_label,patt,patt_label
roast~100_columbian,100% Columbian,roast,Roast
formatsize~12_ct,12 CT,formatsize,FormatSize
formatsize~12_oz,12 oz,formatsize,FormatSize
formatsize~30_oz,30 oz,formatsize,FormatSize
formatsize~48_ct,48 CT,formatsize,FormatSize
subsegment~bag,Bag,subsegment,SubSegment
roast~breakfast,Breakfast,roast,Roast
subsegment~can,Can,subsegment,SubSegment
```

Static Internal RDF Cloud Service Hierarchy Loading Files

These internal RDF Cloud Service hierarchy loading files are static. The GA RDF Cloud Service package contains the hierarchy loading files

Dynamic Internal RDF Cloud Service Hierarchy Loading Files

These internal RDF Cloud Service hierarchy loading files are dynamic. The RDF Cloud Service plug-in generates their hierarchy loading files based on the RDF Cloud Service configuration.

Escalation Levels Hierarchy File

File name: elch.csv.dat

Preprocess Run Hierarchy File

File name: runh.csv.ovr

Preprocessing Path Hierarchy File

File name: path.csv.dat

Long Life Cycle Promotions Hierarchy File

File name: llcp.csv.dat

Short Life Cycle Promotions Hierarchy File

File name: slcp.csv.dat

RHS Hierarchies

The PROR and LOCR internal hierarchies are mirrored hierarchies of the PROD and LOC hierarchies. They are also referred as PROD RHS and LOC RHS. In the RPAS Cloud Edition versions 19.0 and later, PROR and LOCR are considered as virtual hierarchies. Refer to the Oracle Retail Predictive Application Server Cloud Edition Configuration Tools User Guide for information on Virtual Hierarchies.

Since these hierarchies are virtual, you do not have to load the hierarchy files. All of the other operations remain the same. You can register measures on PROR and LOCR and include them in workbooks.

Notes about these virtual hierarchies:

- PROR and LOCR hierarchies have been marked as virtual in the GA configuration.
- We cannot define security dimension on a virtual hierarchy or make them translatable.
- Virtual hierarchies cannot have user defined dimensions.
- If a retailer is upgrading from a pre-19.0 RDF version, then RDF will automatically mark them as virtual and conform to the virtual hierarchy requirements.

RDF Cloud Service Input Data

A detailed data set is required to use the capabilities of RDF Cloud Service to its fullest. Some of the data required is relatively easy to obtain, for example, information about sales. To simplify the data integration, all measure files are configured to be loaded as one measure per file. Each measure's data must be present in a separate file and the file name must be the same as the measure name with the **`.csv.ovr`** extension. All files must be in csv format. During the initial domain build, all data files marked as required are needed with historical data to build the domain.

Measure Name and Intersections

Because many RPAS measure names and intersections are dynamically generated by RDF Cloud Service plug-in. Tokens will be used to represent the RDF Cloud Service level names. The labeled intersection were also listed for measure intersection

Table 2–1 lists the Tokens.

Table 2–1 Token Names

Token Name	Description
_SF_	Short Life Cycle Final Level Name, such as “01”
_CF_	Long Life Cycle Final Level Name, such as “07”
SFS	Short Life Cycle Escalation Level Name, such as “01”, “02” … or “06”
CFS	Long Life Cycle Escalation Level Name, such as “07”, “08” … or “09”
CFP	Long Life Cycle Pooling Level Name, such as “07”, “10” … or “12”
SSG	Short Life Cycle escalation level with Grouping dimension
CSG	Long Life Cycle escalation level with Grouping dimension
CPG	Long Life Cycle pooling level with Grouping dimension
_XP_	Preprocessing path name such as “P01”, “P02”
_BF_	Baseline only LLC final level name such as “07” **Note:** Most measures ending in CF, CFS, or CSG will have the same measure for baseline only levels. The only exception is for the causal related measures. Causal related measures do not exist for baseline only levels
#SLC_data_L_#	SLC final level data intersection = SLC final level intersection – clnd dim + data’s clnd dim generated by plug-in based on user specified plug-in input parameters
#SLC_lvl_L_#	SLC final level timeseries intersection = SLC final level intersection – clnd dim generated by plug-in based on user specified plug-in input parameters
#LLC_frcst_L_#	LLC final level intersection generated by plug-in based on user specified plug-in input parameters
#LLC_frcstTS_L_#	LLC final level timeseries intersection = LLC final level intersection – clnd dim generated by plug-in based on user specified plug-in input parameters
#SLC_seascurve_L_#	SLC level intersection + code generated by plug-in based on user specified plug-in input parameters
#LLC_seascurve_L_#	LLC level intersection – clnddim + woyr generated by plug-in based on user specified plug-in input parameters
#LLC_peff_L_#	LLC promo effect intersection = LLC level intersection –clnd dim + LPRM generated by plug-in based on user specified plug-in input parameters
#SLS_INTX#	Sales History intersection. This labeled intersection is user defined
#SLSNC_INTX#	Sales History intersection -clnd dim This labeled intersection is user defined
#SLC_LVL#	SLC Dashboard final level intersection - clnd dim.
#NIT_ATT_WGT#	Attribute weight intersection, generated by plug-in based on user specified plug-in input parameters
#NIT_SKU_ATT#	Product attribute intersection, generated by plug-in based on user specified plug-in input parameters
#NIT_SKUSTR_INTX#	New Item assignment intersection, generated by plug-in based on user specified plug-in input parameters
#PRESLS_INTX#	LLC Preprocessing data source input intersection

Measure Names and Descriptions

Table 2–2 lists the measure names and descriptions. The measure field descriptions include:

Module Used

This field explains which solution is using the file. The possible values can be: All, Preprocess, New Item, SLC (Short Life Cycle), LLC (Long Life Cycle), Promote

Required or Optional Required

This field means the data is necessary. Optional means that during data load and, if not loaded, certain functionality which uses those measures cannot be used. All administration measures are marked as Optional for data load, since those can be directly set in the Admin workbooks as well.

Load Frequency

This specifies the suggested frequency for the data load. It uses the following values:

- W - Weekly
- A - Anytime as needed or when the values change in source system; it can be weekly, monthly, quarterly, or yearly

Data Source

This specifies the typical data source to get that measure data:

- RI - Oracle Retail Insights or equivalent Data Warehouse solutions
- Admin - Data can be set by Administrator based on customer data referencing sample data in GA domain.
- MFP, IPCS - Oracle Retail Planning Cloud Service or equivalent. Can be readily loaded from RMS or derived from data loaded from RMS.
- ORASE - Oracle Retail Advanced Science. Those are the derived measure files extracted from ORASE integration files.
- RMS - Oracle Retail Merchandising System or equivalent. Can be readily loaded from RMS or derived from data loaded from RMS.
- 3P - Third-party data aggregator such as Nielsen or Symphony IRI.

Load Intersection

Most of the time, the load intersection of the measure is the same as the base intersection of the measure. When the field is empty, the load intersection is the same as base intersection.

Table 2–2 RPAS Measure Names and Intersections

Measure Name	Measure Description	Base Intersection	Measure Type	Module Used	Required or Optional	Load Frequency	Data Source	Load Intersection
rsal	Regular sales	#SLS_INTX#	Real	all	Required	W	RMS/RI	#DAYSLS_INTX#
psal	Promotion Sales	#SLS_INTX#	Real	all	Required	W	RMS/RI	#DAYSLS_INTX#
csal	Clearance Sales	#SLS_INTX#	Real	all	Required	W	RMS/RI	#DAYSLS_INTX#
osal	Other Sales	#SLS_INTX#	Real	all	Optional	W		#DAYSLS_INTX#
flagslc	Short Life Cycle Item Indicator	#slc_lvl#	Boolean	SLC	Required	W		
ldactivefcstitem	Active Forecast Item Indicator	#SLSNC_INTX#	Boolean	all	Optional	W		
PreOosInd	Loaded OutLier Indicator	#PRESLS_INTX#	Boolean	LLC Predemand	Optional	W		
PeOutInd	Loaded Outage Indicator	#PRESLS_INTX#	Boolean	LLC Predemand	Optional	W		
ldPrePpiInd	Loaded Promotion Indicator	#PRESLS_INTX#	Boolean	LLC Predemand	Optional	W		
prdattT	Product Attribute	#NIT_SKU_ATT#	String	Newitem	Optional	W	RMS/RI	
nitdattwgt	Attribute Weight	#NIT_ATT_WGT#	Real	Newitem	Optional	W		
nitfcststovr	New item forecast start date	#NIT_SKUSTR_ INTX#	Date	LLC Newitem	Optional	W		
nisros	New Item Base Rate of Sales	#NIT_SKUSTR_ INTX#	Real	LLC Newitem	Optional	W		
likeitemexmask	Like Item Exclusion Mask	#NIT_SKUSTR_ INTX#	Boolean	LLC New Item	Optional	W		
regprc_SF_	Regular Price	#slc_lvl_L_#	Real	SLC	Require	W		
slsprc_SF_	Sales Price	#slc_data_L_#	Real	SLC	Required	W		
mdind_SF_	Markdown Indicator	#slc_data_L_#	Boolean	SLC	Required	W		
basedmd_SF_	User Provided Base Rate of Sales	#slc_lvl_L_#	Real	SLC	Optional	W		
slcplanstdt_SF_	Item Planned Start date	#slc_lvl_L_#	Date	SLC	Required	W		

Table 2–2 (Cont.) RPAS Measure Names and Intersections

Measure Name	Measure Description	Base Intersection	Measure Type	Module Used	Required or Optional	Load Frequency	Data Source	Load Intersection
slcplanenddt_SF	Item Planned end Date	#slc_lvl_L_#	Date	SLC	Required	W		
pvar_SLCP_	Promotion for Short LifeCycleUser	provided during configuration time	Boolean	SLC	Optional	W		
promoaggprof_ SF_	Promotion Aggregation profile for SLC	User provided during configuration time (Baseline Spread Prof Intx)	Real	SLC	Optional	W		
pvar_LLCP_	Promotion for Long Life Cycle	User provided during configuration time	Boolean/ Real	LLC causal	Optional	W		
bayplan_CF_	Bayesian Plan	#llc_frcst_L_#	Real	LLC	Optional	W		
promoaggprof_ CF_	Promotion Aggregation profile for LLC	User provided during configuration time (Promo Aggprof Intx)	Real	LLC	Optional	W/A		
basespreadprof_ CF_	Baseline spreading profile for LLC	User provided during configuration time (Baseline Spread Prof Intx)	Real	LLC	Optional	W/A		
week53indicator_ CF_	Week53 Indicator	User provided during configuration time	Boolean	LLC	Optional	W/A		
The following measures can be edited in RDF Cloud Service workbooks. They can also be loaded if a data file is provided.								
grpassignPos_ SSG_	TimeSeries Grouping membership for SLC.It shall contain group dimension position names.	#slc_lvl_L_#	String	SLC	Optional	W/A		

Table 2–2 (Cont.) RPAS Measure Names and Intersections

Measure Name	Measure Description	Base Intersection	Measure Type	Module Used	Required or Optional	Load Frequency	Data Source	Load Intersection
grpAssignPos_ CSG_	TimeSeries Grouping membership for LLC. It shall contain group dimension position names.	#llc_frcstTS_L_#	String	LLC	Optional	W/A		
seascureovr_SFS_	User provided SLC seasonal Curve	#slc_seascurve_L_#	Real	SLC	Optional	A		
seabegin_SF_	Season code start. The measure shall contain the position name of WOY dimension (such as WY01). It specify the beginning of item on sale date range	User provided during configuration time (season code intx)	String	SLC	Optional	A		
seaend_SF_	Season code end. The measure shall contain the position name of WOY dimension (such as WY04). It specify the ending of item on sale date range	User provided during configuration time (season code intx)	String	SLC	Optional	A		
sealenmin_SF_	Season length min. It specify the minimum seasonal length of items in a season code.	User provided during configuration time (season code intx)	Integer	SLC	Optional	A		
seaslenmax_SF_	Season length max . It specify the maximum seasonal length of items in a season code.	User provided during configuration time (season code intx)	Integer	SLC	Optional	A		

Table 2–2 (Cont.) RPAS Measure Names and Intersections

Measure Name	Measure Description	Base Intersection	Measure Type	Module Used	Required or Optional	Load Frequency	Data Source	Load Intersection
defescpath_SF_	Default Escalation Path	Elvl+User provided during configuration time (Escalation Path intx)	Integer	SLC	Optional	A		
elasovr_SF_	User Provided Elasticity	#slc_lvl_L_#	Real	SLC	Optional	A		
glescpath_SF_	Global Escalation Path	Elvl	Integer	SLC	Optional	A		
grpAssignPos_CPG_	TimeSeries Grouping membership for LLC Causal Pooling	#llc_frcstTS_L_#	String	LLC	Optional	A		
usrllccurve_CFS_	User Provided LLC Season Curve	#llc_seascurve_L_#	Real	LLC	Optional	A		
week53indicator_CF_	Week53 Indicator	User provided during configuration time (week53 flag intx)	Boolean	LLC	Optional	A		
prmovreff_CFP_	Promotion Effects Override	#llc_peff_L_#	Real	LLC	Optional	A		
defescpath_CF_	Default Escalation Path	Elvl+User provided during configuration time (Escalation Path intx)	Integer	LLC	Optional	A		
glescpath_CF_	Global Escalation Path	Elvl	Integer	LLC	Optional	A		
defpoolesc_CF_	Default Pool Escalation	Elvl+User provided during configuration time (Escalation Path intx)	Integer	LLC	Optional	A		
glpoolesc_CF_	Global Pool Escalation	Elvl	Integer	LLC	Optional	A		

Table 2–2 (Cont.) RPAS Measure Names and Intersections

Measure Name	Measure Description	Base Intersection	Measure Type	Module Used	Required or Optional	Load Frequency	Data Source	Load Intersection
The following measures' data file were generated by RDF Cloud Service plug-in and loaded at domain build/patch time								
promoefftype_CF_	Promotion Model Type for LLC	LPRM	Integer	LLC	Required			
	https://docs.oracle.com/cd/E75759_01/rdf/pdf/cloud/190/html/user_guide/output/est_setup_llc.htm#BABDAEIE							
lprmefftyplist	LLC Promotion Model Type PickList	LPRM	String	LLC	Required			
enabledpromo_SF	Enable SLC Promotions	SPRM	Boolean	LLC	Optional			
promoefftype_SF_	Promotion Model Type for SLC	SPRM	Int	LLC	Optional			
esclist_SF_	SLC Escalation Level Picklist	scalar	String	LLC	Required			
esclist_CF_	LLC Escalation Level picklist	scalar	String	LLC	Required			
poollist_CF_	LLC Pooling Level picklist	scalar	String	LLC	Required			
wblvlrange	Dashboard level range	ELVL	Boolean	Dashboard	Required			
wblvllblmap	Dashboard Level label	ELVL	String	Dashboard	Required			
flvlint	Forecast Level Intersection	ELVL	String	All	Required			
bslpqbfs_BF_	Baseline Position Query	ELVL	Boolean	LLC	Required			
cslpqcfs_CF_	Causal Position Query	ELVL	Boolean	LLC	Required			
cslpqcfp_CF_	Causal Position Query	ELVL	Boolean	LLC	Required			
cslpqcp_CF_	Causal Position Query	ELVL	Boolean	LLC	Required			
slcpqsfs_SF_	SLC Level Position Query	ELVL	Boolean	LLC	Required			

Table 2–2 (Cont.) RPAS Measure Names and Intersections

Measure Name	Measure Description	Base Intersection	Measure Type	Module Used	Required or Optional	Load Frequency	Data Source	Load Intersection
ppsDataSrc_XP_	Preprocessing Input Data Source	Scalar	String	LLC	Required			
ppsOutput_XP_	Preprocessing Output Data Source	Scalar	String	LLC	Required			
ppsMethod_XP_	Preprocessing Methods	RUND	Integer	LLC	Required			
ppsRunLabel_XP_	Preprocessing Run Label	RUND	String	LLC	Required			
ppsFirstAux_XP_	Preprocessing Run Parameter 1	RUND	String	LLC	Required			
ppsSecAux_XP_	Preprocessing Run Parameter 2	RUND	String	LLC	Required			
ppsRunOrder_XP	Preprocessing Run Order	RUND	String	LLC	Required			
ppsRunPreB_XP_	Run Preprocessing Flag	RUND	Boolean	LLC	Required			
ppsStdESAdjust_XP_	Preprocessing Adjustment Mode Flag	RUND	Boolean	LLC	Required			

Historical Data

It is recommended that you have at least two full years of historical data for long life cycle forecasting and one full year of historical data for short life cycle forecasting.

Loading and Extracting Data

Data is loaded into RDF Cloud Service using the Online Administration Tools, which in turn use standard RPAS utilities. For more information on loading and extracting data using Online Administration Tools, see the *Oracle Retail Demand Forecasting Cloud Service Administration Guide*

Loading Image Based Data

RDF Cloud Service is pre-configured to support the display of images for items and product attributes in the Forecast Review and New Item workbooks. Table 2–3 lists the dimension attribute measures used to load images.

Table 2–3 Labeled Intersections

Measure	Hierarchy	Dimension
skuimage	PROD	sku
skupimage	PROD	skup
skugimage	PROD	skug
skurimage	PROR	skur
skprimage	PROR	skpr
skgrimage	PROR	skgr
patvimage	PATR	patv
pattimage	PATR	patt

The Content Server exposes the client's image files placed into a particular directory as HTTP URLs. The images must be defined in the load file in an xml format. The images are available at:

```
http://{content server url}/imgfetch/{sub directory if defined}
```

Sample File for skuimage.csv.ovr

The first field represents the SKU ID followed by the required image location. At a minimum, a thumb size image file must be loaded to show in the pivot table. However, both the thumb and full size images can be loaded.

```
10000010,"<image id=""main"" label=""Front View"">\
<url size=""thumb"">http://msp00alq.us.oracle.com:9001/contentserver/imgfetch/sku_
10000010_main_thumb.jpg</url></image>"
```

Example File for skuimage.csv.ovr

```
10000010,"<image id=""main"" label=""Front View"">\
<url size=""thumb"">http://msp00alq.us.oracle.com:9001/contentserver/imgfetch/sku_
10000010_main_thumb.jpg</url>
<url size=""full"">http://msp00alq.us.oracle.com:9001/contentserver/imgfetch/ sku_
10000010_main_full.jpg</url></image>"
```

Integration

RDF Cloud Service supports the flat file integration of hierarchy and data files from source systems.

Retailers must extract and provide the hierarchy files and data files from their respective source systems as flat files in the required format and upload them to the Oracle Cloud SFTP server ($FTP_INCOMING). The automated process send those files over to the RPAS DB Server and from there the files can be accessed by batch process triggered using the Online Administration Tools. In the same way, exported files in CSV format from the solution are pushed back to the Oracle Cloud SFTP server and from there retailers can download the extracted files.

RDF Cloud Service supports integration with Oracle Retail Merchandising Foundation Cloud Service (RMF CS). If a retailer has RMF CS as the source system for transactional data, they can readily integrate to get foundation hierarchy data and transactional data from RMF Cloud Service. Refer to Appendix B, "Appendix: RDF Cloud Service integration with RMF Cloud Service."

RDF Cloud Service can also integrate with other applications using Planning Data Source (PDS) and Bulk Data Integration (BDI). Refer to Appendix C, "Appendix: RDF Cloud Service Integration with PDS and BDI.".

RDF Cloud Service has the capability to calculate the Demand Transference (DT) Effects and apply it to the forecast if DT is enabled. It needs the Demand Transference (DT) Multiplier as input to calculate these effects. RDF Cloud Service integrates with ORASE to get the DT multipliers. It can also receive the size profiles from ORASE which can be used to spread the Short Life cycle forecast from **SKUP** to **SKU**.Refer to Appendix E for RDF Cloud Service integration with ORASE / RI.

User Roles and Securities

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.

For more information on security, see the *Oracle Retail Predictive Application Server Cloud Edition Administration Guide*. For more information on data security in a cloud environment, see the Hosting Policy documents for the cloud solution.

Internationalization

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

Oracle Retail applications have been internationalized to support multiple languages.

The RPAS platform supports associated solution extensions and solution templates.

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

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

Translations of the solution templates are released. All templates have the ability to support multi-byte characters.

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

Translations are available for RDF Cloud Service for the following languages:

- Chinese (Simplified)
- Chinese (Traditional)
- Croatian
- Dutch
- French
- German
- Greek
- Hungarian
- Italian
- Japanese
- Korean
- Polish
- Portuguese (Brazilian)
- 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 Cloud Edition Administration*.

3

RDF Configuration

RDF is a statistical forecasting solution that uses state-of-the-art modeling techniques to produce high quality forecasts with minimal human intervention.

RDF supports pre-processing, new item/store processing and forecast generation. To obtain good forecast results, the above features need to be configured to work together. RDF is highly configurable and extremely flexible. To streamline RDF implementation and shorten implementation time, several plug-ins are provided to work together with RPAS Configuration Tools. These plug-ins let users input configuration options through the GUI and automatically generate configuration solutions based on the RDF GA master template and user inputs. The configuration solutions generated by the plug-ins are PrepDemand, New Item and RDF. Since Promotions are integral to forecasting, it is part of the RDF solution. The plug-ins auto-generate the hierarchies, measures, rules, workbook templates, taskflow and the Dashboard configuration file that are required by RDF to support the forecasting configuration entered in through the plug-in interface:

Table 3–1 Autogenerated Items from Plug-ins

Autogenerated Entity	Description
Hierarchies	The internal hierarchies required by the solution will be generated by the plug-in. Labeled Intersections are autogenerated.
Measures	All measures necessary to support the base solution will be created.
Rules	All Rule Sets, Rule Groups, and Rules to support the base solution will be created.
Workbook Templates	All pre-defined workbook templates to support the base solution will be created.
Taskflow	The taskflow will be auto generated based on the RDF template and the levels entered in the plug-in.
Dashboard Configuration file	The Dashboard configuration file is auto generated based on the dashboard levels and custom exceptions enter using the plug-in.
Batch Control file	The Batch Control file is auto generated.

RDF Batch Flow Process

Understanding the RDF batch flow process is important before starting RDF Configuration:

RDF has two major batch process:

- RDF Estimation Batch
- RDF Forecast Batch

RDF Estimation Batch

This batch process is suggested to run monthly or quarterly to generate seasonality curve and promotional effects.

This batch process involves the following steps:

1. Running Preprocess Batch
2. Running RDF Estimation

RDF Forecast Batch

It is recommended to run this batch process weekly in order to generate the base demand, update promotion effects and combine base demand, promotional effects and the seasonality curve to produce a forecast.

Note: RDF provides a mechanism to extend the GA batch process. Refer to Customizing the RDF Batch Process.

This batch process include the following steps:

1. Load Weekly Data
2. Running Preprocessing Batch
3. Running New Item Batch
4. Generate System Forecast
5. Adjust System Forecast
6. Approve Adjusted Forecast and System Forecast.

   Forecasts can be approved in three ways:

   - **Manual** - Nothing is approved in the batch process and you must go to the forecast review workbook to approve forecasts.
   - **Automatic** - All forecasts are approved by the system. RDF has defined several GA approval alerts that are available for the approval process.
   - **Approval by Alert** - Approves forecasts based the user specified approve alert. With no alert hit, the forecast is approved. With an alert hit, the forecast is not approved,

     Implementors can also define MyException alerts through the RDF plug-in to create additional approval alerts. These alerts are produced before approval.

> **Note:** There is something tricky here for approval process. If an item/store is approved through either Automatic or Approve by Alerts, then the item/store's alert hit are cleared in all approval alerts including both GA approval alerts and MyException alerts.
>
> That way, when building the Dashboard workbook and the Forecast Review workbook, only the unapproved items still have alerts.

7. Export Forecast

> **Note:** It is recommended to export the forecast as close to the receiving / replenishment system needing the forecast, so as to reflect the current state of the RDF application.

8. Build the Dashboard Workbook

Implementation Process

The RDF GA configuration can be used out of the box to build the RDF domain. The GA configuration has RDF's point of view on the number of final levels, promotions and preprocessing paths.

RDF implementers can modify the RDF GA configuration to meet the retailer's business needs. RDF supports two means to achieve this:

- Configuring the solutions using the plug-ins
- Extensibility of the configuration

This chapter explains how to configure the various solutions using the plug-ins. Extensibility of the configuration is described in Chapter 4, "RDF Cloud Service Extensibility". Although there is a separate plug-in for New Item, Preprocessing and RDF solutions, from the Config Tools UI, we only see two plug-in dialogs – Forecast Common and RDF. This simplifies the configuration process for the implementer.

RDF Implementation Point of View

The RDF GA configuration is RDF's point of view and it is highly recommended to have:

- Two pre-processing paths – one for baseline data source and another for causal data source
- Not more than one Short Lifecycle final level
  - Ten escalation levels or less should be able to cover all business cases
- One Long Lifecycle final level
  - Ten escalation levels or less should be able to cover all business cases
  - Five pooling escalation levels or less should be able to cover all business cases

- Group events into ten promotions or less. By doing this, it is enough promotions to provide granularity yet few enough to provide robustness. Short Lifecycle must have only Boolean promotions. For Long Lifecycle, the price discount must be modeled as exponential
  - Five approval alerts or less. More than five is unwieldy for the user.

Set Up Common Configuration Details

From the Configuration Tools toolbar, select the Automation menu and then from the Forecast Common option, select **Specify Configuration Details**.

Figure 3–1 Configuration Tools: Forecast Common

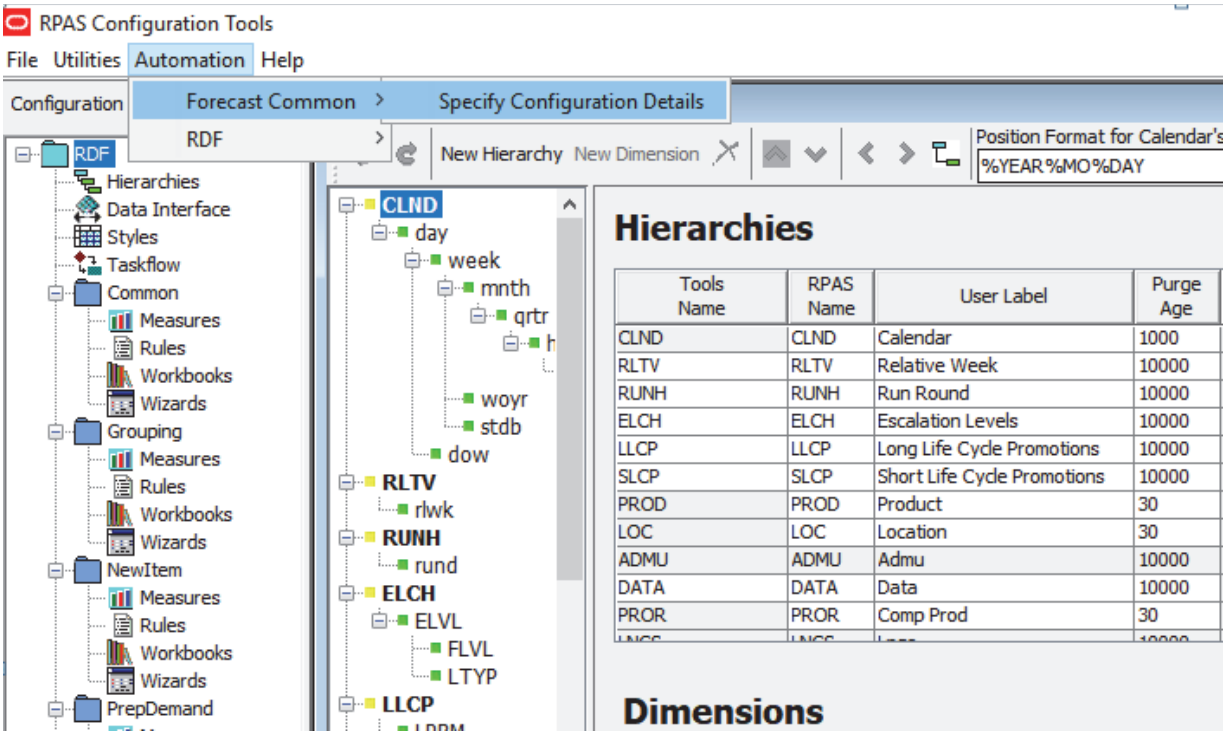
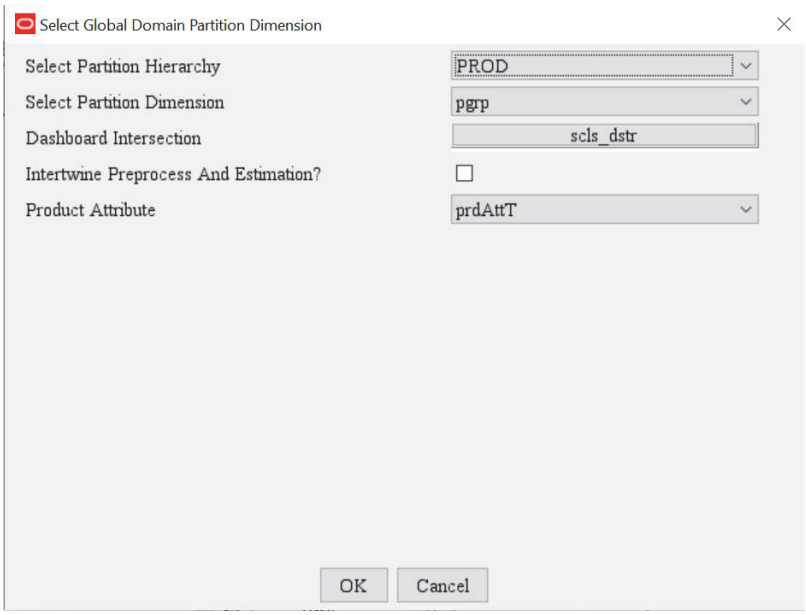


Figure 3–2 Select GLobal Domain Partition Dimension



RDF only supports global domain configuration. So in this step we specify the partition dimension and hierarchy.

Dashboard Intersection is the level at which the information has to be displayed in the dashboard.

The **Intertwine Preprocess and Estimation?** check box allows implementors to indicate if they would like the Preprocess step and Estimation step to be intertwined in estimation batch. This check box should only be switched on when DePromote was selected as a Preprocess method to remove Promotion Effects from original sales. When the check box is turned on, different batch flow logic is followed and different custom hook are provided for batch control files. Implementors need to update the custom batch control files if they would like to check the box and patch the domain. It is recommended to turn this on only for new installation.

The product attribute measure to be used in the RDF and New Item solutions has to be specified in the Common Plug-in. The product attribute measure stores the attribute position name and not the attribute label.

> **Note:** The partition dimension must be specified on the PROD hierarchy. The plug-in validation will ensure this.

Labeled Intersections

The labeled intersections listed in Table 3–2 must be defined before running the RDF plug-ins. The plug-in validation will ensure that the required labeled intersections are defined.

Table 3–2 Labeled Intersections

Labeled Intersection	Definition	Description	Measures Defined
SLS_INTX	sku/stor/week	Sales intersection	pos, rsal, psal, csal, osal
DAYSLS_INTX	sku/stor/day	Sales intersection at day (load intersection)	Can be used as load intersection for sales coming in at day level
PRESLS_INTX	sku/stor/week	Preprocess Sales input/output	llcdata, totadjbasesls, totaladjcalsls
SLSNC_INTX	sku/stor	Sales intersection without calendar	ldactivefcstitem, flagllc
PRESLSNC_INTX	sku/stor/week	Preprocess sales input intersection without calendar	Used by Preprocess indicator measures and parameter measures
PRESLSPROF_INTX	sku/stor/woyr	Seasonal profile intersection	Used by measure to deseasonalize sales during preprocessing

Labeled Intersection Use Cases

Labeled intersections listed in Table 3–2 can be defined based on the retailers business needs. SLS_INTX is the labeled intersection for the incoming sales measures (pos,rsal,psal,csal, osal). PRESLS_INTX is the labeled intersection for the Preprocessing input, output and indicator measures. All the preprocessing indicator measures should match the intersection of the preprocessing input and output measures.

Consider the following two use cases:

- The RDF final level forecast is at sku/chnl/week and the input sales data has to be preprocessed at sku/chnl/week. In this case the PRESLS_INTX can be modified to be at sku/chnl/week. (This would avoid creating any custom measures to match the RDF data source intersection with the preprocessed output measure intersection.)
- The RDF final level forecast is at sku/chnl/week and the input sales has to be preprocessed at sku/stor/week. In this case the PRESLS_INTX can remain at sku/stor/week and all the preprocessing takes place at this level. However we would need a custom measure at sku/chnl/week to serve as RDF data source measure and a custom rule is needed to aggregate from the Preprocessed sales measure (toadjbasesls) to the Custom RDF data source measure.

Common Solutions

Open an RDF GA configuration to see the common modules. This solution should not be modified by the implementer and are considered as non-touch solutions. This solution defines input/output measures for the whole RDF project. The content created in this module will not be modified by the plug-ins. The measures created in these modules are external measures for the plug-ins, and they will serve as inputs to plug-ins. Although this module are not generated by plug-in, It will be overridden in RDF Configuration Automation Script. Any modification by the implementer will be ignored.

Common Solution

In RDF GA, the common solution is used to register measures related to sales and product attribute inputs/outputs to:

- PrepDemand
- New Item
- RDF Solutions

Note: For the common solution, an implementor can only modify the labeled intersection definition that changes the measure intersection in common.

Set up the PrepDemand Solution

The purpose of the PrepDemand solution also referred to as Preprocessing or Data cleansing, is to correct past data points that represent unusual sales values that are not representative of a general demand pattern. Such corrections may be necessary when an item is out-of-stock and cannot be sold, which usually results in low sales. Preprocessing will adjust for stockout for both the current week and the following week because it assumes that the out-of-stock indicators represent end-of-week-stockout. Data Correction may also be necessary in a period when demand is unusually high. The Preprocessing module allows you to automatically make adjustments to the raw POS (Point of Sales) data so that subsequent demand forecasts do not replicate undesired patterns that are caused by lost sales or unusually high demand. Preprocessing can also be used to remove promotion spikes when a promotion indicator is available. Inclusion of a promotion spike can seriously skew the baseline forecasting. It is ideal to remove seasonal effects from sales so that the seasonal pattern does not interfere with the causal estimation, especially when promotion and significant seasonal patterns overlap.

Based on the usage, sales history should be preprocessed in different ways. In RDF GA configuration, two preprocessing paths are configured. Path 01 is used to preprocess sales for baseline forecasting. Path 02 is used to preprocess sales for causal forecasting. For baseline forecasting, the sales history goes through four stages:

- Out-of-stock correction
- Outlier correction
- Promotional spike removal
- Smoothing

For causal forecasting, the sales history goes through three stages: out-of-stock correction, outlier correction and seasonal pattern removal. Each stage is called RUN in preprocessing.

Based on the customer's needs, an implementer can decide how many paths to configure, what kinds of runs to include in each path, and what the input and output measures are for each path. Once the information is fed to the PrepDemand plug-in, the PrepDemand solution will be auto-generated with all necessary measures, rules and workbook templates. Each preprocessing path is also associated with the RDF final level that uses the preprocessed output as RDF data source. This also aids in the calculation of the promo indicator and seasonal profile used during the de-promote and de-seasonalize phases of preprocessing.

Set up the New Item Solution

The New Item module is designed to support the forecast for new item/store. RDF provides three approaches to forecast new item/store:

Forecast Approach	Description
Like Item	The forecast is created based on the forecast of Like Items. The Like Items can be selected manually, and the choices are entered in the User Selected Like Items measure. The task can also be automated if attributes are available. RDF CS then suggests one Like Item in the system recommended Like Item measure. The forecast for the New Item is given by: **`Base demand new item = base demand like item * Adjustment Factor`** The forecast for the New Item is calculated as: **`Forecast at time t = base demand new item * seasonality at time t (coming from escalation level) * promo and price effects (coming from pooling level)`**
Base Rate of Demand	RDF CS calculates the escalated base rate of demand. The forecast for the new item is given by: **`Forecast at time t = base rate of demand (coming from escalation level) * seasonality at time t (coming from escalation level) * promo and price effects (coming from pooling level)`**
User Input	This method is very similar to Base Rate of Demand, with the difference that you have to manually specify a base rate of demand. The forecast is then generated using the same formula as for Base Rate of Demand. **`Forecast at time t = base demand new item * seasonality at time t (coming from escalation level) * promo and price effects (coming from pooling level)`**

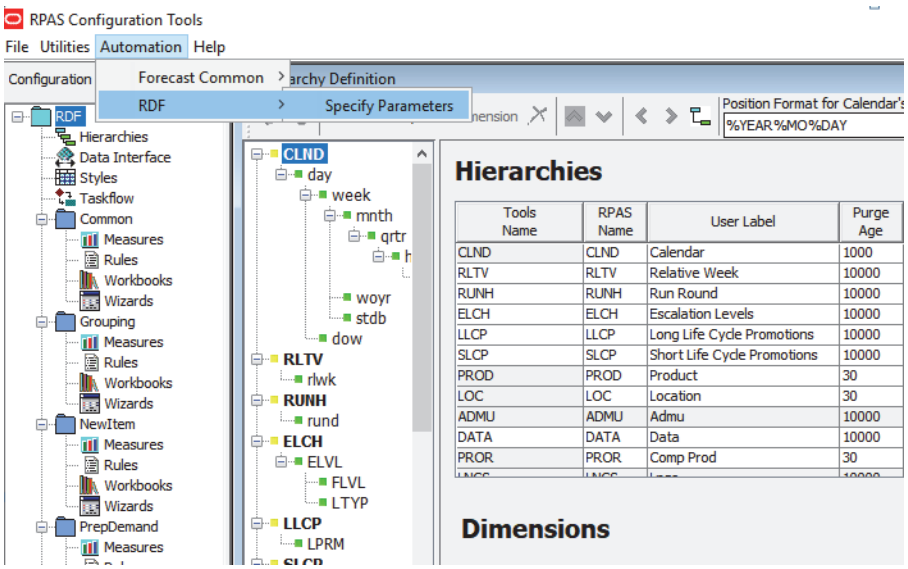
The New Item module provides tools to support the automatic and manual assignment of like item/store to new item/store. If the user can provide product attribute information, the new item can be automatically identified and provided a like item recommendation. If no product attribute information is available, the user has to assign like items manually. New store mapping is always done manually.

Configure New Item Solution

Perform the following steps to generate a New Item solution:

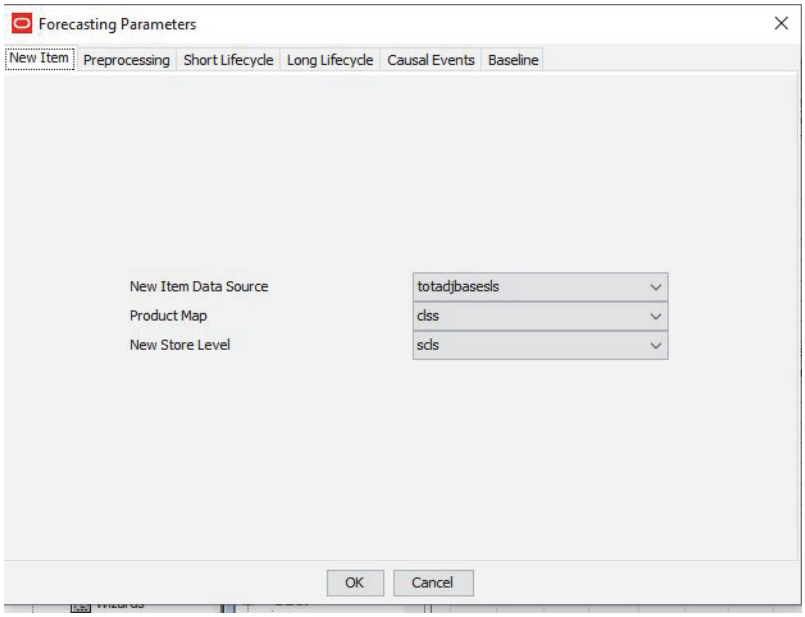
1. From the Configuration Tools toolbar, select the Automation menu and then, from the RDF option, select **Specify Parameters**.

Figure 3–3 Configuration Tools: New Item



2. From the Like Item Parameters utility, specify the properties for the New Item plug-in. Refer to Editing New Item Parameters for details.

Figure 3–4 Like Item Parameters



3. Click **OK** once editing is finished.

Editing New Item Parameters

Table 3–3 lists the New Item parameters available for editing. Ensure that the labeled intersections listed in Table 3–4 are present

Table 3–3 New Item Parameters

Parameter	Description
New Item Data Source	Sales data used to generate forecast for New item/store.
Product Map	This field specifies the range of the like item available to a new item. If the field is populated with clss, it means that only existing items under the same class as the new item are available as like item candidate. The Similarity Score calculation should only be performed between the new item and existing items with in the class.
New Store Level	This field specifies the product level on which like store is assigned to new store. If the field is selected as scls, it means that the like store assignment can be different per subclass.

The following labeled intersections are used to define measures used by the New Item solution. They need to be created and customized before upgrading.

Table 3–4 New Item Labeled Intersections

Labeled Intersection	Dimensions
NIT_ATT	patt
NIT_ATV	patv
NIT_SKU_ATT	sku_patt
NIT_SKU_ATV	sku_patv
NIT_SIM_ATT	sku_skur_patt
NIT_ATT_WGT	clss_patt
NIT_SIM_ATT_PART	pgrp_skur_patt

Preprocessing Configuration Process

The following sections describe the configuration process for preprocessing historical demand.

PreDemand Pre-configuration Data Requirement

There are several parameters within the PrepDemand solution that may reference other measures that are configured external to the solution. Prior to configuring a PrepDemand solution, it is required that these measures already exist within the project:

Measure	Description
Data Source	The PrepDemand plug-in provides a list of existing numeric measures based on the user-specified intersection for a preprocessing path. An implementer selects the measure that stores the input data for preprocessing. This measure should be configured in the Common module.

Measure	Description
Output	The PrepDemand plug-in provides a list of existing numeric measures based on the user-specified intersection for a preprocessing path. An implementer selects the measure that stores the output data from the preprocess. This measure is also normally an input to New Item/RDF plug-in. This measure should be configured in the Common module.
First Aux Measures	For each preprocessing path, the PrepDemand plug-in allows a maximum of six runs, which means six preprocessing stages (one preprocessing method per stage). Each preprocessing method may require supporting measures as inputs. Two supporting measures are allowed for each run. These supporting measures are specified in the fields of First and Second Aux measures. Examples of these measures include out-of-stock indicator or promotional indicator. These measures are external measures to PrepDemand and should be configured in PrepDemand Common.
Second Aux Measures	

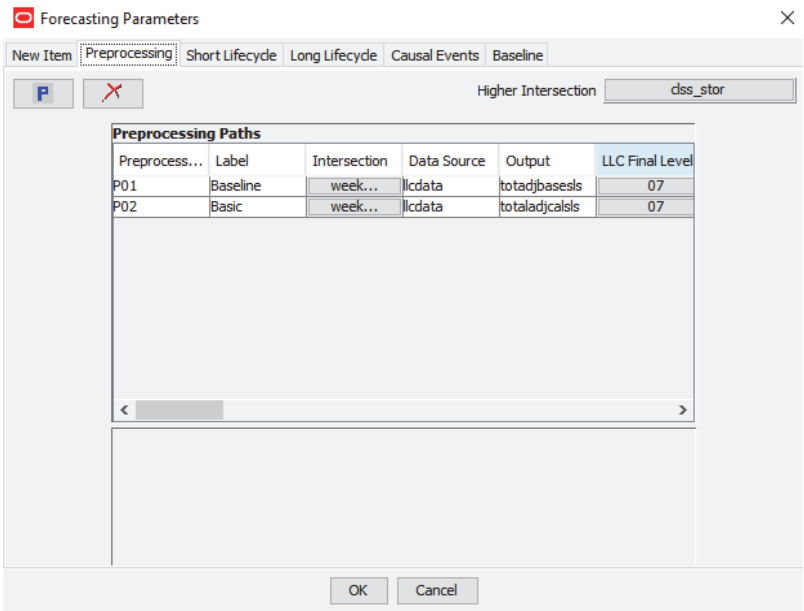
Configuring PrepDemand Solution

Once all input measures are configured, perform the following steps in Configuration Tools and the Preprocessing Parameters utility.

1. From the Configuration Tools toolbar, select the Automation menu and then, from the Prepare Demand option, select **Specify Parameters**.
2. On the Preprocess Parameters utility, click the **P** icon.

   A new preprocessing path is added, and it is assigned the next available level number. To specify the properties for the preprocessing path, see Edit Preprocessing Parameters for details. The Higher Intersection selection box allows you to specify the intersection of default preprocessing parameters.

Figure 3–5 Preprocess Parameters Utility



3. After finishing the configuration, click **OK** to start generating the preprocessing configuration.

Edit Preprocessing Parameters

Table 3–5 lists all of the Preprocessing Parameters.

Table 3–5 Preprocessing Parameters

Preprocessing Path Parameters	Description
Preprocessing Path	The field is the system-assigned path number when a preprocessing path is created. This is a read-only parameter.
Label	The field is the level description that is viewed by the user once the domain is created.
Intersection	The intersection of the preprocessing input and output measures.
Data Source	The data source is the measure to be used as the input data (for example, POS) for the preprocessing.
Output	The output is the measure to store preprocessed result, which may serve as input to New Item /RDF modules.
Method [n]	There are six fields for preprocessing method (method 1 through method 6). The plug-in provides a list to select a specific method each field. Each method is considered a run. The maximum number of runs allowed per path is six.
Run [n] Label	There are six fields to label preprocessing runs. One label per preprocessing method.
First Aux [n]	First Aux and Second Aux are fields to specify supporting measures per preprocessing method, such as seasonal profile, outlier indicator, outage indicator and promotion indicator. For each preprocessing method, the plug-in allows for two optional measures to be used. Some preprocessing methods need only one, others need none. If it is not needed, then leave the field empty. There are six First Aux fields and six Second Aux fields, one per method. Always populate the First Aux field first before using Second Aux. Refer to Table 3–6 for the First Aux and Second Aux supporting measures.
Second Aux [n]	

Table 3–6 lists the supporting measures for the First Aux and Second Aux preprocessing parameters.

Table 3–6 First Aux and Second Aux Supporting Measures.

Method	First Aux	Second Aux
Standard Median	not applicable	not applicable
Oracle Retail Median	not applicable	not applicable
Standard Exponential Smoothing	Outage	event flag
Override	Reference	mask
Increment	Reference	mask
Clear	not applicable	not applicable
Deprice	Price	not applicable
DePromote	promolift	event flag (optional)

Table 3–6 (Cont.) First Aux and Second Aux Supporting Measures.

Method	First Aux	Second Aux
Deseasonal	Seasonal Profile	not applicable

DePromote

This method removes promotion lift from input based on the promolift and eventflag (optional). The promolift measure is expected to be a real measure on the same intersection as input. Its value is 0. Consecutive nonzero promolift periods is considered a promotion window. If all nonzero promolift is 1 for a timeseries, standard exponential smoothing is performed on the input. If the promolift is nonzero (not all ones) and the promotion window is less than or equal to short event max length, Standard Exponential Smoothing is performed on the input. All the parameters related to standard exponential smoothing such as eventflag, history window and future window will apply under this circumstance. When the promotion window is longer than short event max length, the input data is divided by the promolift value to remove promo lift.

This is a hybrid preprocessing method. If short event max length is set to zero, this method always perform Standard Exponential Smoothing. The promolift is used as promotion indicator. If short event max length is set to a larger number such as 9999, output = input/promolift when promolift is non zero. output =input when promolift is zero.

If the DePromote method is used and the promolift measure (prepromolift GA measure) is selected as the promo indicator measure, the same measure must be selected as the event flag (secondary aux measure) for the outage and outlier runs. Basically the even flag measure must be the same measure as the input measure for the de-promote run.

RDF GA has built in logic to calculate promolift based on causal effects and promotion calendars. There is a boolean measure named PreCalcPromolift in the Batch Flow Management workbook. When PreCalcPromolift is set to true, promolift for a certain preprocess path is automatically calculated based on its default forecast level's causal effects and promotion calendars. When PreCalcPromoLift is set to false, then promolift can be an custom measure that is loaded.

> **Note:** The Calculate Promotion Lift (PreCalcPromoLift) and Calculate Promotion Indicator (precalcpromoind) are mutually exclusive and should not be selected together.

In the common plug-in, the **Intertwine Preprocess and Estimation?** check box is introduced specifically for the DePromote method. There are two different flows for estimation batch when DePromote is configured. The different flow produces different results. When the check box is turned off, the estimation batch follows the normal process with preprocessing completed first and followed by estimation of seasonality curve and promotional effects second. The following is a description of the batch steps when preprocessing and estimation are not interwined.

When a domain is freshly built, there is no causal effects calculated. RDF GA calculates an initial promolift as 1 when merged promotion indicator is on and 0 when the merged promotion indicator is off.

After the first time estimation batch is run, then the following steps are performed:

1. If any input data reset is enabled for a preprocess path, clear out any preprocess adjustments, seasonal curves, and promotional effects. Save the Preprocess window.
2. Calculate the preprocessing indicators including promolift and seasonal profile.

   If seasonal curves were not populated, 1 is used as default for initial seasonal profile for preprocessing. The initial promlift is 1.0 when the merged promotion calendar is on.
3. Run Preprocessing.
4. Determine if the Estimation needs to run twice. Rerun Estimation a second time if the promotion effects or seasonal curves do not exist.
5. Run Estimation batch to get causal effects and seasonal curve.
6. If Estimation needs to run a second time, then update preprocessing indicators based on the result from Step 5.

   Calculate updated promolift using new effects and promotion calendars.

   Calculate Seasonal Profile based on new seasonal curves.
7. If Estimation needs to run a second time, then run Preprocessing again
8. If Estimation needs to run a second time, then run Estimation Batch again to generate causal effects and seasonal curves.
9. Perform the following:
   - Restore the Preprocessing window
   - Set the reset preprocess data to false.
   - Set the run estimation twice to false.

When causal effects or seasonal curves exist, then the Estimation Batch has been run in the domain at least once before.

If the Estimation Batch is set to run again, then both preprocessing and estimation will only run once.

Only Steps 2-5 are performed. The remaining steps are skipped.

When the **Intertwine Preprocess and Estimation?** check box is selected, the estimation batch is performed using the following steps:

1. Wipe out previous sales adjustment and causal effects.
2. Calculate historical promotion lift based on causal effects and promotional calendar. Because the causal effects were wiped out, the weekly promotion lift were all 1s.
3. Run the preprocessing path that is configured to generate baseline estimation data source. It uses the DePromote method with promotion lift calculated from Step 2. Because the promotion lift were all ***1***s, the preprocessing special expression automatically switched to Standard Exponentially Smooth to remove the promotion lifts.
4. Calculate approved seasonal curves (**`appllccurve_CF_`**) using preprocessed sales from Step 3.
5. Calculate seasonal profile (**`preseasprof`**) using escalation path and approved seasonal curves from Step 4.

6. Run the preprocessing path that is configured to generate data source for causal. It uses the seasonal profile from Step 5 to deseasonalize the original sales.
7. Calculate approved promo effects (**`prmappeff_CFP_`**) using deseasonalized sales from Step 6.
8. Calculate weekly promotion lift using approved effects from Step 7 and promotion calendars. This time, the promotion lift can have numbers other than ***1*** and ***0***.
9. Run the preprocessing path that is configured to generate baseline estimation data source. It uses the DePromote method with promotion lift calculated from Step 8.
10. Calculate approved seasonal curves (**`appllccurve_CF_`**) using preprocessed sales from Step 9.
11. Calculate seasonal profile (**`preseasprof`**) using escalation path and approved seasonal curves from Step 10.
12. Run the preprocessing path that is configured to generate data source for causal. It uses the seasonal profile from Step 11 to deseasonalize the original sales.
13. Calculate approved promo effects (**`prmappeff_CFP_`**) using deseasonalized sales from Step 12.

There is a difference between the two modes of running the estimation batch. The non-interwine mode is an iterative mode. If the causal effects converge quickly, it will stabilize in a few runs. If it does not converge, the effects tend to be different after each run.

The interwine mode always starts with no effects and performs interpolate first and DePromote by promotion lift the second time. It will generate the same result every time. It is more consistent. There different custom hook for these two mode and it is not simple to switch between these two modes. Refer to Chapter 4, "RDF Cloud Service Extensibility," for the custom hooks that are available.

Deseasonal

This method removes the seasonality from the input based on the seasonal profile generated during each run of the estimation.

Increment

This method increments or decrements the destination measure by the source measure, which is adjusted by the adjustment percentage according to the mask. It is recommended for updating outliers or data gaps when an existing reference measure exists as a default adjustment.

Increment provides the following features:

- It is a simple data increment of a given percentage of the reference data to copy from.
- Has one required parameter, Reference measure to increment by.
- This has to be specified as the first auxillary measure in the preprocess plugin tab.
- It may or may not take outage information (for example, event) as an input to mask the operation.
- This can be specified as an optional second auxillary measure in the preprocess plugin tab.
- Can accept another optional parameter, Ratio of reference to actually increment by.
- This can be specified in the Delta parameter measure in the Preprocess workbook.

Override

This method overrides the destination measure with the source measure that is adjusted by the adjustment percentage according to the mask. It is recommended for filling data gaps when an existing reference measure exists as a default value.

Override provides the following features:

- It is a simple data copy of a given percentage of the reference data to copy from.
- Has one required parameter, Reference measure to copy data from.
- This has to be specified as the first auxiliary measure in the preprocess plug-in tab.
- It may or may not take outage information (for example, event) as an input to mask the operation.
- This can be specified as an optional second auxiliary measure in the preprocess plug-in tab.
- Can accept another optional parameter, Ratio of reference to actually copy.
- This can be specified in the Delta parameter measure in the Preprocess workbook.

Deleting a Preprocessing Path

Deleting a preprocessing path causes the system-assigned enumerated values in the path name to renumber such that paths are in consecutive order, starting with preprocessing path 01. Deleting a preprocessing path may impact any solution configuration that uses a specific preprocessing output.

> **Caution:** If the domain using the configuration has previously been installed, there is potential to lose data associated with a path that has been deleted or renumbered.

Perform the following steps to delete a preprocessing path:

1. On the Preprocessing Parameters utility, highlight the number of the path that you want to delete from the path window.
2. Click the **X** icon to delete the path. The path is deleted.
3. Select **OK** to regenerate the solution with the changes to the PrepDemand configuration.

Edit PrepDemand GA Configuration

The PrepDemand autogeneration process creates all hierarchy dimensions, measures, rules and workbook template to support the essential PrepDemand functionality.

> **Note:** It is recommended to leave the plug-in generated configuration alone and not to modify it manually.

PrepDemand Plug-in Validation

RDF validates the PrepDemand plug-in:

- Each Preprocess path must have at least one run.
- For each run, a preprocess method must be selected.
- The preprocess input and output measures must be on the same intersection.

Configuring the RDF Solution

In RDF, the Demand Model to generate the forecast is:

*Demand = Base Demand * Seasonality * Promo Effects * Price Effects*

This is the basic model used to forecast short lifecycle and long lifecycle items. However the approach to calculate each of these components might differ.

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, calculating the seasonality curves and effects at a higher level in the hierarchy based on an escalation path, would generate a reliable forecast. The RDF plug-in provides a mechanism to define the final levels and escalation levels; and the associated parameters for each level. The default escalation path is the order in which the escalation levels are defined in the plug-in. This can be edited in the setup workbooks from the UI. Users can also override the escalation path at the final level intersection from the UI.

The RDF solution can be configured using the following tabs in the RDF plug-in UI:

Table 3–7 RDF plug-in UI Tabs

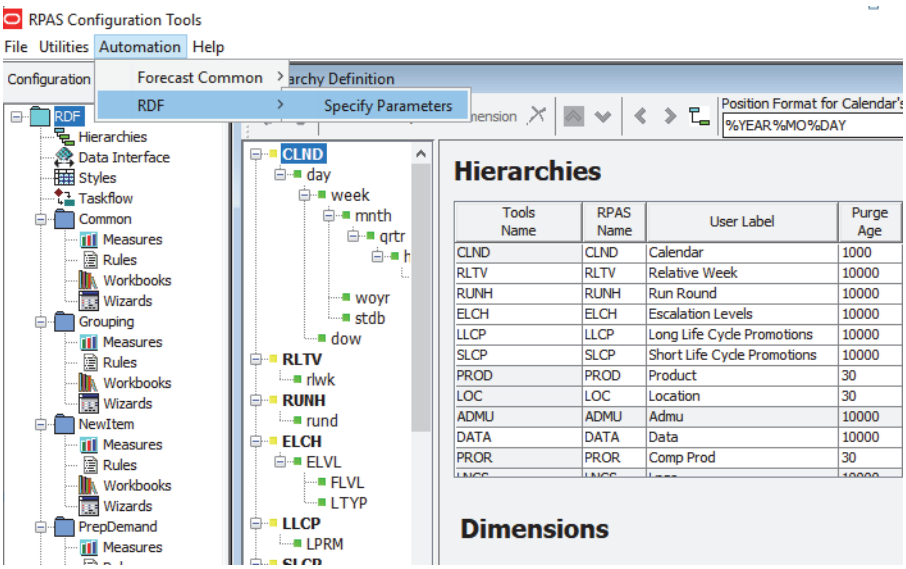
Tabs	Description
Short Lifecycle	Define and configure Short Lifecycle final level and escalation levels
Long Lifecycle	Define and configure Long Lifecycle final level and escalation levels. These final levels generate a causal forecast by applying the casual effects on top of the baseline. Level, trend, seasonality curve and promotion effects can be estimated on these levels. The forecast is derived based on these factors.
Causal Events	Define and configure the causal events for Short Life cycle and Long Lifecycle items.
Baseline	Define and configure Long Lifecycle final level and escalation levels. This generates a baseline forecast. This can be used when we only have base demand and seasonality curve, but do not have promo effects or price effects.

Generate RDF Solutions

Perform the following steps to generate an RDF solution:

1. From the Configuration Tools toolbar, select the Automation menu and then, from the RDF option, select **Specify Parameters**. The following steps outline the process for configuring RDF forecast levels.

Figure 3–6 Configuration Tools: RDF



2. Select the tabs to configure the various parameters for Short Lifecycle, Long Lifecycle, and Causal Events.Each tab is discussed in detail in subsequent sections.

3. Configure a forecast level:

   a. To configure a final forecast level:

   From the Forecasting Parameters utility, click the F icon. A new final level is added, and it is assigned the next available level number. Specify the properties for the final level. See Editing Forecast Level Parameters for details.

   b. To configure an Escalation level:

   From the Forecasting Parameters utility, highlight the final level number in which the new source level will be associated from the Level window and then click the **S** icon instead. A new escalation level is added, and it is assigned the next available number. Specify the properties for the escalation level.

   c. To configure a Pooling level (applicable only for Long Lifecycle tab):

   From the Forecasting Parameters utility, highlight the final level number in which the new pooling level will be associated from the Level window and then click the **P** icon instead. A new pooling level is added, and it is assigned the next available number. Specify the properties for the source level.

Note: To remove a final level, escalation level or pooling level, select the forecast level and then click the **X** icon. Deleting a final level will remove all of its associated escalation and pooling levels.

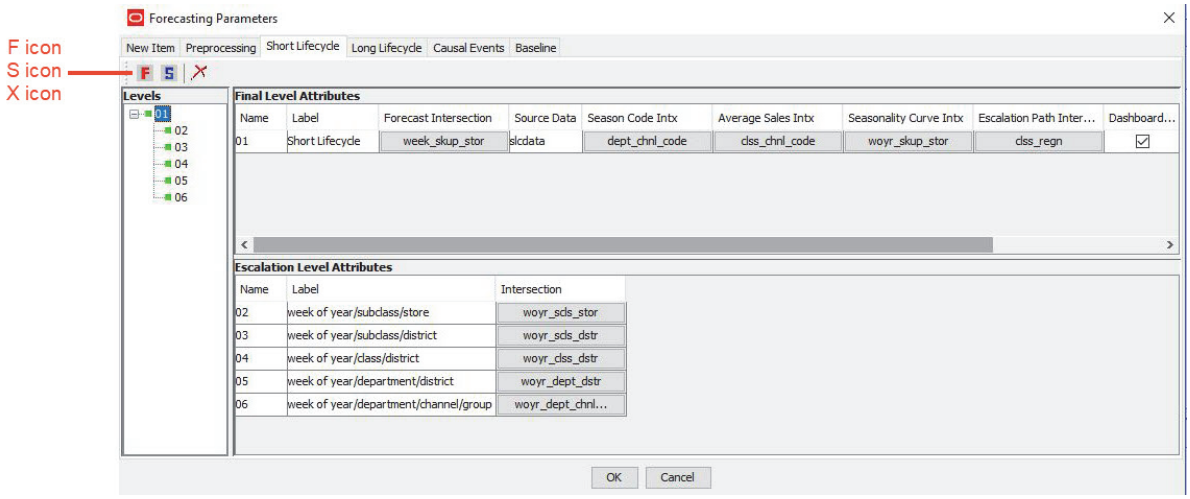
Note: A causal final level should have at least one pooling level. For best performance and accuracy the pooling level should be higher than the final level.

Configure Short Lifecycle Forecast Parameters

Perform the following steps to configure Short Lifecycle Forecast parameters:

1. Select the Short Lifecycle tab in the RDF plug-in UI.
2. Specify the parameters for the short lifecycle levels. See Edit Short Lifecycle Forecast Level Parameters.

Figure 3–7 Short Lifecycle Forecast Parameters



Edit Short Lifecycle Forecast Level Parameters

Table 3–8 lists all of the Short Lifecycle Forecast Level Parameters.

Table 3–8 Short Lifecycle Forecast Level Parameters

Short Life Cycle 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. Level labels may not exceed 40 characters. A hyphen '-' should not be used before or after the Forecast Level label. An example of a Forecast Level label that would violate this requirement is: -1:itm/str/week - Final-. This example is acceptable as: 1-itm/str/week – Final A colon ':' should not be used at all in the Level label. An example of a Level label that would violate this requirement is 1: itm/str/week-
Intersection	The intersection is the hierarchy dimensions that define the forecasting and escalation levels.To configure an escalation level supporting grouping, grouping dimension needs to be included in the intersection of the escalation level. For example, **`Dept/DVSN/grpd/woy`** is grouping escalation level intersection. The prod/loc under a particular dept/dvsn intersection were grouped according certain criteria. Seasonal curves were created is going to be on dept/dvsn/grpd/woy.

Table 3–8 (Cont.) Short Lifecycle Forecast Level Parameters

Short Life Cycle Parameters	Description
Forecast Intersection	This defines the level at which forecast will be generated.
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 list are populated with all non-string and non-Boolean type measures that are configured in the project.
Season Code Intersection	Short Lifecycle items can be grouped together based on their selling pattern and assigned to a season code in the UI. In the plug-in we specify the intersection at which the season code is defined. In GA config, it is defined at dept/chnl/code.
Average Sales Intersection	For Short Lifecycle, to calculate the price effects, we transform the sales to remove seasonality and only keep markdown effects. During this transformation, the sales are averaged at the level defined by the "Average Sales Intersection" parameter. In GA config, it is defined at clss/chnl/code.
Seasonality Curve Intersection	This defines the level of the seasonality curve that will be used for forecasting. In GA config, it is defined at woyr/skup/stor.
Escalation Path Intersection	This is the level at which the default escalation path is defined. In GA config, it is at clss/regn.
Dashboard Level	This parameter indicates that this final level has to be seen in the dashboard. Ideally we would have only one short lifecycle level. But if an implementer chooses to have more than one short lifecycle levels, there can be only one short lifecycle dashboard level.
Promo Agg Profile Intersection	This defines the Aggregation Profile intersection for promotions.
My Exception	This parameter provides a mechanism for the implementer to configure custom real time alerts. These alerts will be used during the batch for Forecast approvals and also seen in the dashboard exception profile as a separate tile. The implementer can enter the labels for the alert and the secondary measure such as variance measure. Note that the implementer is responsible to configure the rule/rule group (based on the alert definition) to populate the boolean measure (and variance measure) for the My Exception alert. After the plug-in automation has run, the implementer also needs to verify the alert condition settings and that the alert is on the intended worksheet. If an item/store is approved through either automatic or approve by alerts, the item/store's alert hit were cleared in all approval alerts including both GA approval alerts and MyException alerts. That way, when building the Dashboard workbook and Forecast Review workbook, only the unapproved items still have alerts.

Short Lifecycle Plug-in Validation

RDF validates the Short Lifecycle plug-in:

- All the escalation levels must be defined higher than the final Forecast level intersection.
- The seasonality curve intersection must be on PROD/LOC/woyr.

- The Season Code Intersection must be on PROD/LOC/SeasonCode. It must be higher than or equal to final level intersection.
- The Average Sales intersection must be within the season code intersection and partition dimension and higher than or equal to final level intersection.

Configure Long Lifecycle Forecast Parameters

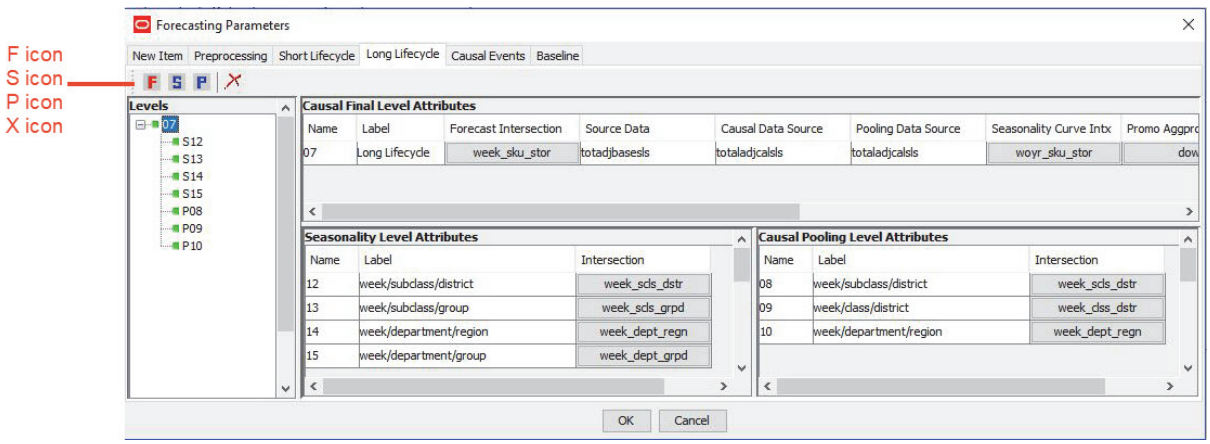
The Long Lifecycle tab in the RDF plug-in dialog enables us to define long life cycle final levels and escalation levels. This is most common approach to be used when promo effects and price effects are available. The final level forecast generated takes into account seasonality and casual effects. In this approach the baseline forecast is internally generated and the effects are applied on top of it. In this tab we define three type of levels:

- Final Level – This is the final forecast level.
- Escalation Levels – defines the escalation levels used to calculate the seasonality curve
- Pooling Levels – defines the pooling levels used to calculate the casual effects.

Perform the following steps to configure Long Lifecycle Forecast parameters:

1. Select the Long Lifecycle tab in the RDF plug-in UI.
2. Specify the parameters for the long lifecycle levels. See Edit Long Lifecycle Forecast Level Parameters.

Figure 3–8 Long Lifecycle Forecast Parameters



Edit Long Lifecycle Forecast Level Parameters

Table 3–9 lists all of the Long Lifecycle Forecast Level parameters.

Table 3–9 Long Lifecycle Forecast Level Parameters

Long Life Cycle 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.

Table 3–9 (Cont.) Long Lifecycle Forecast Level Parameters

Long Life Cycle Parameters	Description
Level Label	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. A hyphen '-' should not be used before or after the Forecast Level label. An example of a Forecast Level label that would violate this requirement is: -1:itm/str/week - Final-. This example is acceptable as: 1-itm/str/week – Final A colon ':' should not be used at all in the Level label. An example of a Level label that would violate this requirement is 1: itm/str/week-
Intersection	The intersection is the hierarchy dimensions that define the forecasting and escalation levels.To configure an escalation level supporting grouping, grouping dimension needs to be included in the intersection of the escalation level. For example, **`Dept/DVSN/grpd/woy`** is grouping escalation level intersection. The prod/loc under a particular dept/dvsn intersection were grouped according certain criteria. Seasonal curves were created is going to be on dept/dvsn/grpd/woy.
Forecast Intersection	This defines the level at which forecast will be generated. Usually the final level forecast intersection will be at week. In case of daily causal final level, the effects will be calculated at week level and the forecast will be spread from week to day.
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 list are populated with all non-string and non-Boolean type measures that are configured in the project.
Casual Data Source	Here we select the name of the data source measure used to calculate promotion effects at the final level. The values in this list are populated with all non-string and non-Boolean type measures that are configured in the project.
Pooling Data Source	Here we select the name of the data source measure used to calculate promotion effects at the pooling levels. The measure may be different than the causal data source, because at the pooling levels the data source can be normalized, something that is not recommended.
Promo Agg Profile Intersection	This defines the intersection of the Promo Aggregation Profile measure. It is used only for Daily Promotions, to aggregate promotions defined at day up to the week.
Baseline Spread Profile	This defines the intersection of the Baseline Spread profile measure. It is used only for Daily Causal to spread the baseline forecast from week to day level.
DT Level	This determines if the level enables the demand transference effects or not. If enabled, the DT corresponding measures will be created during automation.
Seasonality Curve Intersection	This defines the level of the seasonality curve that will be used for forecasting. In GA config, it is defined at woyr/skup/stor.
Escalation Path Intersection	This is the level at which the default escalation path is defined. In GA config, it is at clss/regn. It is used to defined the intersection of Intermediate Level views and forecast intermediate level parameters, as well as the lowest level of the wizards in Estimation Setup and Forecast Setup workspaces.

Table 3–9 (Cont.) Long Lifecycle Forecast Level Parameters

Long Life Cycle Parameters	Description
Dashboard Level	This parameter indicates that this final level has to be seen in the dashboard. Ideally we would have only one Long lifecycle level. But if an implementer chooses to have more than one Long lifecycle levels, there can be only one Long lifecycle dashboard level.
Extra Week Indicator	This defines the level at which the extra week indicator measure (week53indicator) is registered.
Max Horizon	Defines the maximum number of weeks of forecast length.
My Exception	This parameter provides a mechanism for the implementer to configure custom real time alerts. These alerts will be used during the batch for Forecast approvals and also seen in the dashboard exception profile as a separate tile. The implementer can enter the labels for the alert and the secondary measure such as variance measure. Note that the implementer is responsible to configure the rule/rule group (based on the alert definition) to populate the boolean measure (and variance measure) for the My Exception alert. After the plug-in automation has run, the implementer also needs to verify the alert condition settings and that the alert is on the intended worksheet. If an item/store is approved through either automatic or approve by alerts, the item/store's alert hit were cleared in all approval alerts including both GA approval alerts and MyException alerts. That way, when building the Dashboard workbook and Forecast Review workbook, only the unapproved items still have alerts.

Long Lifecycle Plug-in Validation

RDF validates the Long Lifecycle plug-in:

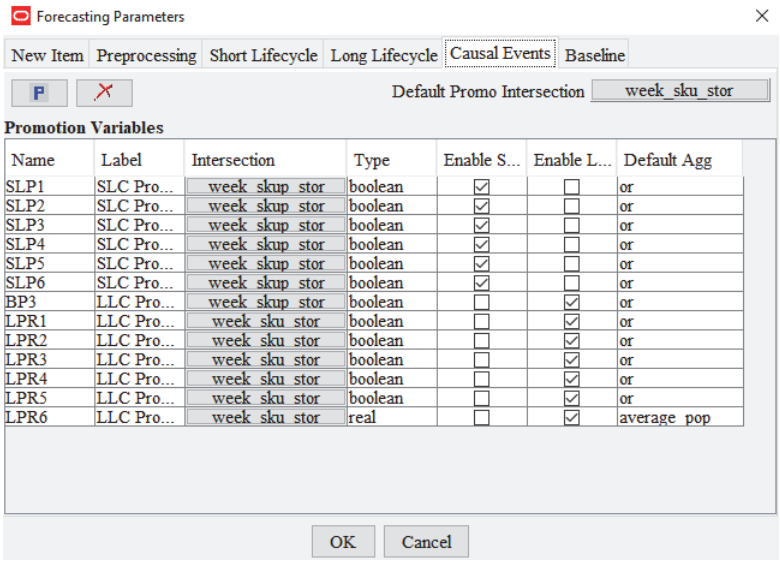
- Forecast and escalation intersections cannot be on alternate product hierarchy.
- Forecast intersection must either be at week or day.
- Escalation path cannot be on alternate Product hierarchy.

Configure the Parameters for Causal Events

This tab provides a mechanism to setup promotional and causal events, such as radio advertisements and holiday occurrences.

1. Select the Causal Events tab in the RDF plug-in UI.
2. Specify the parameters for the casual events. See Edit Parameters for Causal Events.

Figure 3–9 Parameters for Causal Events



Create a Promotion

Perform the following steps to create a promotion:

1. From the Promote Parameters utility, click the **P** icon.

   A new promotion is added, and it is assigned a default promotion number for the Promotion Name (for example, P001).

2. Specify the properties for the promotion. See Edit Parameters for Causal Events.

Deleting a Promotion

Perform the following steps to delete a promotion:

1. From the Promote Parameters utility, highlight the promotion that you want to delete from the configuration.
2. Click the **X** icon. The promotion is deleted.
3. Click **OK** to regenerate the solution with the changes to the promotion configuration.
4. Patch the domain with the new configuration.

Edit Parameters for Causal Events

Table 3–9 lists all of the Casual Events parameters.

Table 3–10 Casual Events Parameters

Causal Events Parameters	Description
Default Intersection	The Default Intersection is the intersection at which any new promotion will be defined. Editing the Default Intersection will not affect any existing promotions.

Table 3–10 (Cont.) Casual Events Parameters

Causal Events Parameters	Description
Promotion Name	The Promotion Name is the internal system identifier of the promotion. The system will initially assign a generic Promotion Name (P001), but this value may be overwritten. The Promotion Name may not be greater than four characters. The following characters may not precede or follow the name that is entered in this field: '( )' Example: (xmas) '-' Example: -xmas- The following must not be used at all in the Promotion Name: ':' Example: xmas:
Promotion Label	The Promotion Label is the description of the promotion that will be viewed by the user once the domain is created. Promotion Labels may not exceed 40 characters. The following characters may not precede or follow the label that is entered in this field: '( )' Example: (xmas) '-' Example: -xmas- The following must not be used at all in the Promotion Name: ':' Example: xmas:
Promotion Intersection	The Promotion Intersection is the hierarchy dimension that defines the promotion. It is pre-populated with the value set in the Default Intersection at the time when the promotion is created.
Type	The Type is the data type of the promotion variable. Promotion Variables may be defined as Boolean or Real types. The value in this parameter defaults to Boolean.
Enable LLC	Enables the promotion for Long Lifecycle.
Enable SLC	Enables the promotion for Short Lifecycle.
Default Agg	The default aggregation to be used for the display of causal events.

Casual Events Plug-in Validation

RDF validates the Casual Events plug-in:

- Boolean promotions can only be of Linear model type.
- Promotions of real data type can be exponential or power model type.
- Short Lifecycle promotions can only be of boolean type.

Configure Baseline Only Long Lifecycle Forecast Level Parameters

The Baseline tab although not recommended, can be used when there are no promo effects or price effects to feed into the demand model. In such cases, we have only have the base demand and seasonality curve. Hence we only have the baseline final level and the escalation levels to calculate the seasonality curve.

1. Select the Baseline tab in the RDF plug-in UI
2. Specify the parameters for the baseline only long lifecycle levels. See Edit Baseline Only Long Lifecycle Forecast Level Parameters.

Edit Baseline Only Long Lifecycle Forecast Level Parameters

Table 3–11 lists all of the Baseline Only Long Lifecycle Forecast Level parameters.

Table 3–11 Baseline Only Long Lifecycle Forecast Level Parameters

Baseline Only LLC 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. Level labels may not exceed 40 characters. A hyphen '-' should not be used before or after the Forecast Level label. An example of a Forecast Level label that would violate this requirement is: -1:itm/str/week - Final-. This example is acceptable as: 1-itm/str/week – Final A colon ':' should not be used at all in the Level label. An example of a Level label that would violate this requirement is 1: itm/str/week-
Intersection	The intersection is the hierarchy dimensions that define the forecasting and escalation levels.To configure an escalation level supporting grouping, the grouping dimension needs to be included in the intersection of the escalation level. An example of a grouping escalation level intersection is:Dept/DVSN/grpd/woy. The prod/loc under a particular dept/dvsn intersection were grouped according certain criteria. Seasonal curves that were created will be on dept/dvsn/grpd/woy.
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 list are populated with all non-string and non-Boolean type measures that are configured in the project.
Seasonality Curve Intersection	This defines the level of the seasonality curve that will be used for forecasting. In GA config, it is defined at week/skup/stor.
Escalation Path Intersection	This is the level at which the default escalation path is defined. In GA config, it is at clss/regn.
Dashboard Level	This parameter indicates that this final level has to be seen in the dashboard. Ideally we would have only one Long lifecycle level. But if an implementer chooses to have more than one Long lifecycle levels, there can be only one Long lifecycle dashboard level.
Extra Week indicator	This defines the level at which the extra week indicator measure (extra***) is registered.
Max Horizon	Defines the maximum number of weeks of forecast length.

Table 3–11 (Cont.) Baseline Only Long Lifecycle Forecast Level Parameters

Baseline Only LLC Parameters	Description
My Exception	This parameter provides a mechanism for the implementer to configure custom real time alerts. These alerts will be used during the batch for Forecast approvals and also seen in the dashboard exception profile as a separate tile. The implementer can enter the labels for the alert and the secondary measure such as variance measure. Note that the implementer is responsible to configure the rule/rule group (based on the alert definition) to populate the boolean measure (and variance measure) for the My Exception alert. After the plug-in automation has run, the implementer also needs to verify the alert condition settings and that the alert is on the intended worksheet. If an item/store is approved through either automatic or approve by alerts, the item/store's alert hit were cleared in all approval alerts including both GA approval alerts and MyException alerts. That way, when building the Dashboard workbook and Forecast Review workbook, only the unapproved items still have alerts.

RPAS Rolling Calendar

RDF CS has enabled an RPAS Rolling Calendar feature for the Long Lifecycle and Short Lifecycle Forecast Review workbook templates. Refer to the *Oracle Retail Predictive Application Server Cloud Edition Configuration Tools User Guide* for more details.

In essence this allows us to define a calendar window based on RPAS_TODAY. The main use case for this feature is for auto workbook builds, where in the calendar window advances based on RPAS_TODAY.

Figure 3–10 Rolling Calendar

Out of the four rolling calendar range measures, only the Minimum Future has been set to the Alert Calculation window. This is set during the forecast batch.

The Minimum Past defaults to 0, which means it is not required to pull in any week prior to TODAY.

The Maximum Past and Maximum Future are also not set and will default to the current Calendar pre-range.

Note: The rolling calendar feature is not extensible and implementors cannot edit the rolling calendar range measures.

Translation Process in RDF

As part of the domain build or patch process, RDF loads the GA translation files (which includes RPAS and taskflow files).

RDF then loads any custom translations that you may have placed on the SFTP server in the INCOMING_FTP_PATH/translation directory.

During the patch, RDF also loads previously uploaded translation files.

For details, refer to the *Internationalization* chapter in the *Oracle Retail Predictive Application Server Cloud Edition Administration Guide*.

Note: As part of configuration or extensibility, if the implementer changes the labels of the RDF level or Preprocessing paths in the plug-in, RDF generates the corresponding English (and non-english_us) translations in the **`r_msglabel`** measure and loads it.

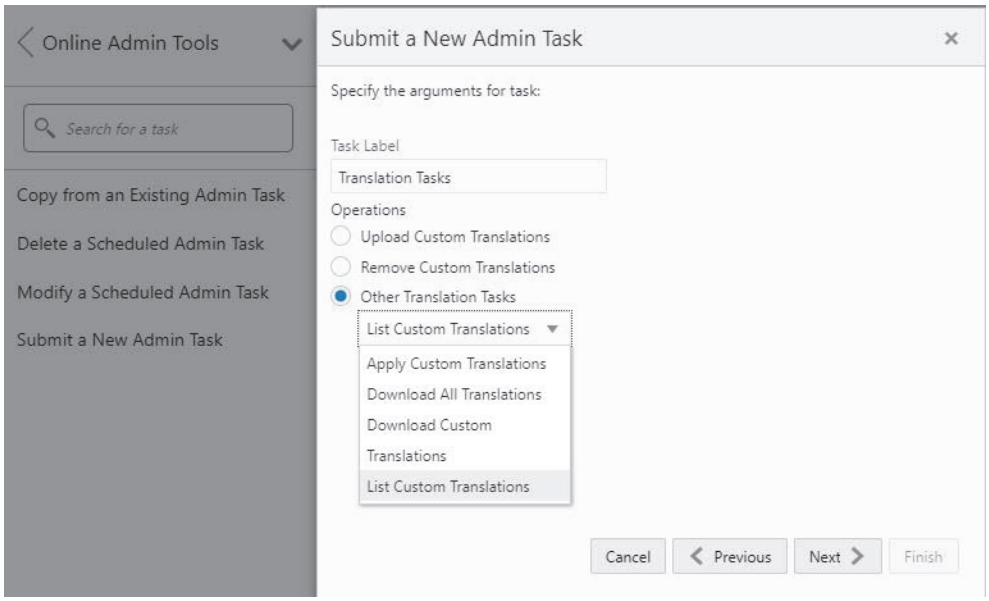
For locale specific translations, it is the implementer's responsibility to upload the correct translation files.

Since the RDF level labels are appended to the worksheet labels, the implementer should upload the new labels.

Perform the following steps to access the position names to create the locale specific file:

1. Make sure the browser locale is English - United States
2. From the OAT configure batch task, go to the **Translation Task** and then, **Download All Translations**.
3. The **`r_msglabel.csv.ovr`** file contains the English labels as updated by the implementer in the RDF plug-in.
4. This file can serve as an example to create the locale specific **`r_msglabel`** file.
5. The locale specific file can contain only the records with the updated labels.
6. Revert the browser language to the original locale.
7. Upload the locale specific **`r_msglabel.csv.ovr`** file using the **Translation Task** from the **OAT Configure Batch Task**.

Figure 3–11 Translation Tasks



4

RDF Cloud Service Extensibility

As mentioned in Chapter 3, "RDF Configuration," apart from configuring the RDF application through the plug-ins, RDF also supports extensibility of the GA configuration. This chapter describes the rules and restrictions enforced to extend the RDF GA configuration, so as to preserve the customizations in future patch and upgrades.

RDF also provides a mechanism for implementers to extend the RDF Batch process, where in custom rule groups can be executed during the batch process. RDF also supports Dashboard extensibility.

The solutions within the RDF GA configuration can be categorized into:

- Solutions that is extensible.
- Solutions that cannot be extensible.

The Solutions that cannot be customized are the ones not generated by the plug-in. The non extensible solution is:

- Common

Generally the solutions generated by the plug-in can be customized; however they should follow the rules for extending a solution. The extensible solutions are NewItem, PrepDemand and RDF.

Supported Customization of RDF Configuration

The following sections list the customizations that are allowed to the RDF configuration. These configuration components can be customized:

- Solution
- Measures
- Rules and Rule groups
- Workbooks and worksheets
- Hierarchy
- Taskflow
- Styles

All the names of custom realized measure, rule set, rule group, rule, workbook, worksheet, and styles should begin with the prefix **c_** or **C_**.

Custom worksheets can only be added into existing RDFCS GA workbook tabs for the plug-in generated solutions.

Rules for Customizing Hierarchy

The following hierarchy customizations are allowed to the RDF configuration:

- Clients are allowed to add new hierarchy or add new dimension into the existing hierarchy. No dimension can be added to calendar hierarchy that is lower than day. No change can be made to the RDF internal hierarchies.
- Clients are allowed to change the label of existing hierarchies or dimensions.
- All the dimension and roll-up order in the product, RHS product, location and RHS location hierarchy must be preserved in the custom configuration.

Rules for Adding Measures

The following rules apply when adding measures to the RDF configuration:

- Clients are allowed to add new custom measures into the custom solution and reference them as an external measure in the extensible solutions.
- Clients can also add new custom metric as a major component in the extensible solutions. It is strongly recommended not to mix custom metrics with the RDF metrics.
- Custom measures should follow the naming convention and should begin with a 'C_' or 'c_' prefix.
- Only the published GA measures can be used in custom rules and custom workbooks. Only writable GA measures can be used on the left hand side of an rule expression. The read only GA measures can only be used on the right hand side of the rule expression.

Publishing Measures

The published GA measures can be divided into these categories:

- **Read only**—can only be used on the right hand side of the expression
- **Writable**—can be used on both the left hand side and right hand side of the expression
- **RuleGroupOnlyWritable**—a specific measure that can be read/write in the specified rule group
- **Loadable**—measures that can be loaded using OAT and can be present in the custom load batch control file.
- **WorkbookMeasureOverride**—measures whose property can be overridden in the associated workbook
- **ReadableExecutionSet**—list of GA batch control exec set names that can be called from within custom batch control exec file.

The list of published measures can change based on forecast levels in a particular configuration. Therefore it is dynamically generated at each RDF configuration regeneration.

The contents of the list is saved in a file named: **`publishedRDFMeasures.properties`**.

The file is located under **`[config]/plugins`**. Before writing custom rules, regenerate your RDF configuration and then open the file to search for published RDF measures.

Custom Measure Characteristics:

- Each line of the file has three fields that is `|` separated.

- The first field is either Read Only, Writable, or RuleGroupOnlyWritable.
- The second field is the measure name.
- The third field is the measure label.
- For RuleGroupOnlyWritable, it includes an extra field, *rule group name*, in the last column.
- For WorkbookMeasureOverride, the last column is the name of the workbook in which this measure is allowed to be overridden.

Example 4–1 Sample Custom Measure

```
ReadOnly|PreSeaProf|Seasonal Profile
ReadOnly|activefcstitem01|Active Forecast Items
ReadOnly|activefcstitem07|Active Forecast Items
```

Generally, forecasting parameter overrides such as Forecast Method Override, Custom Alerts, auxiliary inputs to RDF such as Promotion Aggregation Profile, and Grouping Membership were writable because an implementer may set them up through customized rules.

Rules for Adding Custom Rules

The following rules apply when adding custom rules to the RDF configuration:

- Custom rule sets, rule groups and rule names should begin with the 'C_' or 'c_' prefix.
- Custom rule groups should not include any GA rules.
- Custom rules can use the published readonly GA measures listed in the publishedRDFMeasures.properties file. However, the custom rules cannot modify the value of the readonly GA measure. Hence the readonly GA measure cannot appear on the LHS of a custom rule.
- Custom Rules can be added to custom rule group. It can also be added to the plug-in generated GA workbook rulegroups such as load rule group, calc rule group, refresh rule group, commit rule group and custom menu rule. However Custom Rules can not be added to plug-in generated batch rule group.

Rules for Workbooks and Worksheets Extensibility

The following rules apply when adding custom rules to the RDF workbooks and worksheets extensibility:

- New Custom workbook and worksheets names should begin with the 'C_' or 'c_' prefix.
- Apart from the Custom Solution, custom workbooks can also be added to the extensible RDF GA solutions.

Workbook Measure Override Extensibility

RDF supports certain GA measures to be overridden in the GA workbook. These measures are listed in the WorkbookMeasureOverride section of the published**`RdfMeasures.properties file`**.

For example:

```
WorkbookMeasureOverride|ppsstdesadjustp01|Std ES Adjustment Flag|PpsAdminP01
```

This indicates that the measure **ppsstdesadjustp01** can be overridden in the PpsAdminP01 workbook.

The following rules apply to override measure properties:

- Base State and Agg State can be overridden.
- Range property of static picklists can be overridden. Note that options can only be removed; new options cannot be added.

Elapsed Lock Override

RDF supports the RPAS platform feature of Elapsed Lock Override in the following scenarios:

- Custom measures in a workbook can have the Elapsed Lock Override set to true.
- Custom workbooks can have this field set to true for GA measures that are in the Writable list of the published measures.

> **Note:** If a GA measure is not been enabled as Elapsed Lock Override, the following steps can achieve the same behavior:
>
> 1. Make sure the GA measure is writable.
> 2. Register a custom measure and load it from the GA measure.
> 3. Set the custom measure as Elapsed Lock Override.
> 4. Edit the custom measure in the workbook.
> 5. Commit the custom measure back into the GA measure.

Rules for Adding Custom Real Time Alerts into Existing Workbooks

Perform the following steps when adding custom real time alerts into existing workbooks.

> **Note:** These steps have to be performed using RPAS Configuration Tools. Copying, pasting or direct editing of xml files is prohibited.

1. To add custom real time alert into existing workbooks, all measures related to the custom real time alert need to be added to the workbook.
2. Create a style for the custom real time alert in the configuration.
3. Create a custom real time alert in an RDF workbook using the measures and style created from the previous steps.
4. If a real time alert defined in custom solution will be used in a GA workbook, the real time alert measure should be imported as an external measure in the corresponding GA solution.
5. We must ensure that the rule group consistency is maintained while adding any custom rules that might be needed to calculate an alert measure.

The RDF plug-in will preserve a custom real time alert during regeneration

Adding a Custom Solution

Custom solution is a separate solution within the RDF Configuration. It can be used to accommodate custom workbooks, rules, alerts to do custom reporting, custom logic

and threshold alerts by using GA measures (based on the extensible GA measures in Table 4–1). In addition, measures and alerts defined in the custom solution can be plugged into existing workbooks in GA solution based on the contexts defined. Clients are allowed to create their own custom solutions by following the rules mentioned above. To use a GA measure in custom workbooks, the GA measure should be imported as an external measure in custom solution.

Adding Custom Styles

New styles can be added in the Style Definition window of Configuration Tools. The custom style name should be prefixed with either c_ or C_. Style names that do not adhere to the naming convention will be caught during the configuration validation. Any new style added will be retained during upgrades and patches.

Validating the Customized Configuration

A script, **rdf_config_validation.ksh**, has been provided to allow the customer or implementer to validate that the customizations conform to the rules outlined above. For details of the script, refer to Configuration Validation.

This script can be run on Windows with the RDF Starter Kit. To do this, the implementer will need to make sure that they have a pristine copy of the GA configuration as well as the custom configuration.

For example, if the GA configuration has been copied to **C:\Oracle\configurations\GA\RDF** and the custom configuration is in **C:\Oracle\configurations\RDF**, then the script can be called from a Cygwin **zsh** shell:

```
$RPAS_HOME/bin/rdf_config_validation.ksh -n RDF -d
/cygdrive/c/Oracle/configurations -c
/cygdrive/c/Oracle/configurations/GA/RDF/RDF.xml
```

Successful Run of the Validation Script

If all the validations pass, it will output the following message:

Example 4–2 Message for Successful Run of Validation Script

```
09:04:47 : INFORMATION : rdf_config_validation.ksh[0] - rdf_config_validation.ksh
completed.
09:04:47 : INFORMATION : rdf_config_validation.ksh[0] - Program completed
successfully.
09:04:47 : INFORMATION : rdf_config_validation.ksh[0] - Exiting script with code:
0
```

Unsuccessful Run of the Validation Script

If all the validations do not pass, it will output the following message:

> **Note:** The bold line shows where the details of the validation failure are in the log. (In the actual log, this line is not bold.)

Example 4–3 Message for Unsuccessful Run of Validation Script

```
09:15:12 : INFORMATION : rdf_config_validation.ksh[0] - For details of validation,
look in '/cygdrive/d/retek/logs/2017-07-18/rdf_config_validation.091506.1/rdf_
config_validation.log'.
```

```
09:15:12 : INFORMATION : rdf_config_validation.ksh[0] - _call executing command
'execplug-inTask.sh
RDF:com.retek.labs.rdf.plug-in.installer.RDFConfigurationValidation
/cygdrive/c/Oracle/configurations/GA/RDF/RDF.xml /cygdrive/c/Oracle/configurations
RDF'
09:15:17 : INFORMATION : rdf_config_validation.ksh[0] - _call of command
'execplug-inTask.sh
RDF:com.retek.labs.rdf.plug-in.installer.RDFConfigurationValidation
/cygdrive/c/Oracle/configurations/GA/RDF/RDF.xml /cygdrive/c/Oracle/configurations
RDF' complete
09:15:17 : ERROR : rdf_config_validation.ksh[0] - Nonzero exit status code.
09:15:17 : INFORMATION : rdf_config_validation.ksh[0] - Exiting script with code:
9
```

Taskflow Extensibility

The RDF Taskflow is extensible. The implementer can customize the taskflow in Configuration tools to add custom taskflow components like activitives, tasks, steps, tabs and worksheets. Any custom taskflow component added to GA taskflow component will be retained after plug-in automation. As part of extensibility, RDF provides a mechanism where in, the implementer can hide certain components of the GA configuration and taskflow by editing a property file. The property file is a simple text file named **extend_rdf.properties** and is located inside the plug-in directory of the configuration.A sample file is included in the plug-ins directory of the GA configuration for reference.

For example, **RDF\plug-ins\ extend_rdf.properties**

The format of the file is shown as:

Stage|Component|Action|Value

For example, **Customization | Worksheet | Hide | activity_ni.task_ niattmaint.NITREVSht1**

Each line consists of four fields separated by the **'|'** character. The value field can contain a comma separated list of values. Note that the value field should specify the fully qualified name of the taskflow component. Refer to the sample file. Any line that begins with a **'#'** character is considered a comment line and is ignored.

The names of the Taskflow entities can be found in the **taskflow.xml** file located in the configuration directory.

The various GA configuration components that can be hidden are listed in the following table:

Component	Description
Activity	Hides the specified Taskflow activity. The value field is the taskflow activity name.
Task	Hides the specified Taskflow task. The value field is the taskflow task name.
Step	Hides the specified Taskflow step. The value field is the taskflow step name.
Tab	Hides the specified Taskflow tab. The value field is the taskflow tab name.
Worksheet	Hides the specified worksheet. The value field is the worksheet name.

Component	Description
Realtime Alert	Hides the specified Real Time Alert. The value field is the real time alert name.

Customizing the RDF Batch Process

This section describes how to customize the RDF GA batch process to meet the business needs of the retailer. Details on the RDF GA batch process is described in the Administration guide. The Configured Batch tasks have the below tasks related to batch control:

- Retrieve Batch Control File – allows the current batch control files to be retrieved for inspection and modification.
- Update Batch Control File – After inspecting the current batch control files, the implementer can edit the batch control files to customize the batch process.

Details on the previous two tasks are described in the *Oracle Retail Demand Forecasting Administration Guide*.

The RDF Batch process is based on the RPAS Enterprise Edition Batch Framework, which makes use of a set of control files. Table 4–1 lists the RDF Batch control files that can be customized. For detailed information on the RPAS Batch Framework, refer to the *Oracle Retail Predictive Application Server Implementation Guide*.

Table 4–1 Customizable RDF Batch Control Files

Control File	Description
batch_exec_list.txt	This is the controller and entry point for all the other services, specifying groups of services to be run in a specific order.
batch_calc_list.txt	This control file groups all the calc services that need to run using mace.
batch_refresh_list.txt	This control file groups all Workbook refresh rule groups
batch_loadmeas_list.txt	This control file groups measures that need to be loaded into domain using the measure load service
batch_exportmeas_list.txt	This control file groups measures that need to be exported out of the domain using export measure service.
batch_xform_list.txt	This control file handles the transform file service to perform file transformations to support simple integration capabilities.
batch_oat_list.txt	This file lists the configured batch tasks that appear in the OAT drop down list.

Custom Hooks and Boolean Scalar Measures for Flow Control

There are two ways to customize the batch control files:

- Custom Hooks
- Boolean Scalar Measures for Flow Control

The custom hooks are an optional batch set executed by GA batch control files. The implementor can define the contents of these batch set in the customized batch control file that lives in the [domain]/batch_control_cust. If these hooks are not defined, the batch process skips these hooks, If they are defined, its contents are executed.

RDF also defines a list of Boolean Scalar Measures in the domain to control if certain GA defined batch sets can be skipped or not. The following tables lists the hooks and Boolean Scalar Measures.

Estimation Phase

Table 4–2 lists the hooks for the Estimation Phase.

Table 4–2 Estimation Phase Hooks

Hook	Description
hook_pre_slc_est_SF_	This hook is added before the GA Short Lifecycle estimation calculations. _SF_ needs to be replaced by level number.
hook_post_slc_est_SF_	This hook is added after the GA Short Lifecycle estimation calculations. _SF_ needs to be replaced by level number.
hook_pre_llc_est_BF_	This hook is added before the GA Long Lifecycle (baseline only) estimation calculations. _BF_ needs to be replaced by level number.
hook_post_llc_est_BF	This hook is added after the GA Long Lifecycle (baseline only) estimation calculations. . _BF_ needs to be replaced by level number.
hook_pre_csl_est_CF_	This hook is added before the GA Long Lifecycle (casual) estimation calculations. . _CF_ needs to be replaced by level number.
hook_post_csl_est_CF_	This hook is added after the GA Long Lifecycle (casual) estimation calculations. . _CF_ needs to be replaced by level number.

Preprocess is Intertwined with Estimation Phase

Table 4–3 lists the hooks for when preprocess is intertwined with the Estimation Phase.

Table 4–3 Preprocess is Intertwined with the Estimation Phase Hooks

Hook	Description
hook_pre_preprocess_seasonal_CF_	Where_**CF**_ should be replaced by the final level name. This hook is called before running preprocessing path to produce baseline data source
hook_post_preprocess_seasonal_CF_	Where_**CF**_ should be replaced by final level name. This hook is called after running preprocessing path to produce baseline data source
*hook_pre_preprocess_promo_CF_	Where_**CF**_ should be replaced by final level name. This hook is called before running preprocessing path to produce causal data source
*hook_post_preprocess_promo_CF_	Where_**CF**_ should be replaced by final level name. This hook is called after running preprocessing path to produce causal data source
hook_post_csl_estimate_depromote_CF_	Where_**CF**_ should be replaced by final level name. This hook is called after the estimation batch is done when depromote preprocessing is configured.
hook_ppsindicator_CFSEASONALXP_	Where_**CFSEASONALXP**_ should be replaced by the preprocessing path name. The path should be the one that produce baseline data source. This hook is called after GA promotion indicators and promo lift calculation and before calling preprocessing special expression_. It is intended for implementer to add custom steps to calculate promolift if they set precalcpromolift to false.

Table 4–3 (Cont.) Preprocess is Intertwined with the Estimation Phase Hooks

Hook	Description
hook_post_csl_est_seasonal_CF_	Where_**CF**_ should be replaced by final level name. This hook is called after creating approved seasonal curves.
hook_ppsindicator_CFPROMOXP_	Where_**CFPROMOXP**_ should be replaced by preprocessing path name. The path should be the one that have deseasonal configured and produce causal data source. This hook is called after GA calculate seasonal profile and before running the preprocessing special expression. It is intended for implementer to add custom steps to modify seasonal profile.
hook_post_csl_est_promo_CF_	Where_**CF**_ should be replaced by final level name. This hook is called after producing approved promo effects.

Weekly Batch Forecasting Phase

Table 4–4 lists the hooks for the Weekly Batch Forecasting Phase.

Table 4–4 Weekly Batch Forecasting Phase Hooks

Hook	Description
hook_preload	This hook is before data loading, at the begining of batch weekly forecast
hook_pre_post_data_load	This hook is between GA measure load and post_data_load rule group run
hook_pre_preprocess	This hook is between post_data_load rule group run and preprocess
hook_ppsindicator	This hook is between preprocess seasonal profile calculation and preprocessing batch
hook_post_preprocess	This hook is after the preprocessing phase and before generating the forecasts.
hook_pre_forecast	This hook is after New Item calculation and before the forecast generation step.
hook_frcst_adjust_SF_	This hook is provided to add custom forecast adjustment calculations for Short Lifecycle and Long Lifecycle. Thiss hook is before the alert calculation step. _SF_,_BF_ ,_CF_ needs to be replaced by a level number.
hook_frcst_adjust_BF_	
hook_frcst_adjust_CF_	
hook_frcst_alert_SF_	This hook is provided to add custom forecast alert calculations for Short Lifecycle and Long Lifecycle. This hook is before the Forecast Approval step. _SF_, _BF_ and _CF_ needs to be replaced by a level number.
hook_frcst_alert_BF_	
hook_frcst_alert_CF_	
hook_frcst_approval_SF_	This hook is provided to add custom Forecast approval calculations for short Lifecycle and Long Lifecycle. This hook is added before the dashboard calculations. _SF_, _BF_ and _CF_ needs to be replaced by a level number.
hook_frcst_approval_BF_	
hook_frcst_approval_CF_	
hook_post_forecast	This hook is between forecast and export
hook_post_export	This hook is after export

Boolean Scalar Measures

Table 4–5 lists the Boolean Scalar Measures.

Table 4–5 Boolean Scalar Measures

Boolean Scalar Measure	Description
loadrmsdata	This measure is defaulted to false. Set it to true if RDF is integrated with RMS.
runpreprocess	This measure is defaulted to true. Set it to false if preprocessing is not configed or user would like to skip preprocess for batch_ weekly.
PreCalcPromoInd	This measure is defaulted to true. RDF automatically calculate the promotion indicator used in preprocessing through merging of promotion variables in causal forecast level. Set it to false if customer would like to load the promotion indicator for de-promote instead of merging promotion variables.
runnewitembatch	This measure is defaulted to true. Set it to false if new item is not configed or user would like to skip new item batch for batch_weekly.
runestimate_SF_	This measure is defaulted to true. Set it to false if customer would like to avoid running estimation on certain short life cycle levels. _SF_ needs to be replaced by level number.
runfrcst_SF_	This measure is defaulted to true. Set it to false if customer would like to avoid running forecast on certain short life cycle level. _SF_ needs to be replaced by level number.
runestimate_BF_	This measure is defaulted to true. Set it to false if customer would like to avoid running estimation on certain baseline only level. _BF_ needs to be replaced by level number.
clearcurve_BF_	This measure is defaulted to true. Set it to false if customer would like to keep seasonal curves for previous estimation run if no season curve is produced from current run. _BF_ needs to be replaced by level number.
runfrcst_BF_	This measure is defaulted to true. Set it to false if customer would like to avoid running forecast on certain baseline only level. _BF_ needs to be replaced by level number.
runnewitem_BF_	This measure is defaulted to true. Set it to false if customer would like to avoid incorporate new item forecast on certain baseline only level. _BF_ needs to be replaced by level number.
runestimate_CF_	This measure is defaulted to true. Set it to false if customer would like to avoid running estimation on certain causal level. _ CF_ needs to be replaced by level number.
clearcurve_CF_	This measure is defaulted to true. Set it to false if customer would like to keep seasonal curves for previous estimation run if no season curve is produced from current run. _CF_ needs to be replaced by level number.
runfrcst_CF_	This measure is defaulted to true. Set it to false if customer would like to avoid running forecast on certain causal level. _ CF_ needs to be replaced by level number.
runnewitem_CF_	This measure is defaulted to true. Set it to false if customer would like to avoid incorporate new item forecast on certain causal level. _CF_ needs to be replaced by level number.

RDF Batch Control File Customization Guidelines

Follow these guidelines for RDF Batch Control File Customization:

- The file, **batch_oat_list.txt**, is the only batch control file in which customers can overwrite the GA setnames (such as exec, calc).
- For all other batch control files, avoid overwriting GA setnames. GA batch control files have provided various hooks for the batch process. For additional custom steps, try to put them into the hooks.
- GA batch control files have provided a mechanism to skip certain GA steps using boolean scalar measure that can be set in the domain. For example loadrmsdata will allow skip of rms data transformation. runpreprocess, runnewitembatch and runexport allows skip of preprocess, newitem batch and GA export steps. To skip the GA steps, use these mechanism instead of overwriting GA setnames.
- For GA hierarchy that is unused in your implementation such as attribute hierarchy, just provide empty hierarchy file. For unused GA measures,no need to provide the data file. RPAS would be able to skip it if no files were provided.
- Do not remove any GA clnd hierarchy reorder step, this step is very important for proper functioning of RDF.
- For ease of maintenance, all custom batch set name or step names should be prefixed with **c_**

Examples

The following is an example of custom batch_exec_list.txt, batch_calc_list.txt, batch_loadmeas_list.txt and batch_exportmeas_list.txt.

In this example, the following modification were added to batch _weekly process.

- Hierarchy and measure data file were unpacked.
- Custom measures were loaded after GA measure load.
- Outlier indicator for preprocessing were calculated use custom rules
- Custom approval alerts were run afetr GA alerts and before approval
- Promotion effects were exported after GA exports

Batch Control Samples

The following sections list samples of batch control processes.

batch_exec_list.txt

Example 4–4 # unpack data file before data load

```
hook_pre_load | unpack      | rdf_hier.tar.gz
hook_pre_load | unpack      | rdf_meas.tar.gz
```

Example 4–5 # load custome measures after GA hier and measure load

```
hook_pre_post_data_load | measload    | c_weeklyLoad
```

Example 4–6 # calculate outlier indicator used in preprocess using custom rules

```
hook_ppsindicator | calc | c_outlier_calc
```

Example 4–7 # calculate custom approval alerts after GA approval alerts

```
hook_frcst_alert07 | exec | c_calc_cust_alerts
```

Example 4–8 # custom export

```
hook_post_export | measexport | c_export_promoeffects
c_calc_cust_alerts | calc |c_custalert1
c_calc_cust_alerts | calc |c_custalert2
```

batch_calc_list.txt

Example 4–9 #outlier calculation

```
c_outlier_calc | G | GROUP | c_HBICalcTodayIdx
c_outlier_calc | G | GROUP | c_dataprocess
c_outlier_calc | G | GROUP | c_calc_outlier
```

Example 4–10 #custom approval alerts calculation

```
c_custalert1 | G | GROUP | c_custalert1
c_custalert2 | G | GROUP | c_custalert2
```

batch_loadmeas_list.txt

Example 4–11 # load custom measure

```
c_weeklyLoad | M | c_ActiveItem
c_weeklyLoad | M | c_DisContinue
```

batch_exportmeas_list.txt

Example 4–12 # export custom measure

```
c_export_promoeffects|O|promoeffects.csv.dat
c_export_promoeffects|X|storsku_lprm
c_export_promoeffects|F|c_ExportMask
c_export_promoeffects|S|ftp
c_export_promoeffects|M|prmbldeff07
```

Custom Batch Control Validation

The extensible / custom batch control files need to follow the guidelines previously listed so as to future proof the retailer. That means the retailer should receive software updates without breaking the existing customizations. To ensure that the batch control file guidelines are adhered to, a batch control validation module has been added.

The **`rdf_config_validation`** script has an optional parameter **`-b`** <parent directory of batch control files> which will validate the batch control files.

Batch control validation rules:

- Apart from the **`batch_oat`**_list, none of the set names in the other batch control files can be overridden. That is, GA set names cannot be used in custom batch control files.
- None of the custom batch control files can call the GA set names.
- The **`batch_calc_list`** can only specify custom rule group names. Cannot specify expressions and GA rule group names.

- The **batch_loadmeas**_list can specify measures that are listed in the Loadable or Writable list of the published measures in the published**RdfMeasures.properties** file
- The **batch_exportmeas**_list can specify measures that are listed in the ReadOnly or Writable list of the published measures in the publishedRdfMeasures.properties file.
- All custom set names should have a prefix of **c_.**

Note that the batch control validation is called automatically during domain build or patch. It is also called when the batch control files are uploaded using the Upload Batch Control files from OAT.

Parallel Processing Control for Cross-domain Causal Estimation

The calculation for cross-domain causal estimation requires a lot of memory. It is the only calculation in the RDF batch that may require implementors to set a maximum parallel process based on their domain configuration and hardware. An implementor can change the cross domain causal estimation thread count measure in the Batch Flow Management workbook to specify the maximum number of thread count allowed for this calculation. If the measure value is zero, it is using the same default settings as other calculations.

Dashboard Extensibility

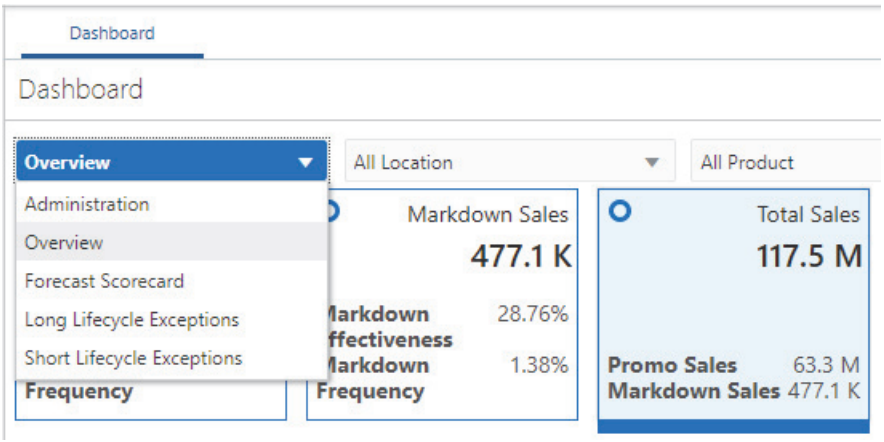
RDF supports Dashboard Extensibility by allowing the Dashboard Settings configuration file to be customized.Refer to the chapter, "Configuring Dashboards in RPASCE EE" in the *Oracle Retail Predictive Application Server Cloud Edition (RPASCE) Configuration Tools User Guide* for detailed information on Dashboard components.

As part of extensible dashboard, the following are supported:

- Adding custom Metric and Exception profiles.
- Adding a custom tile to GA Metric and Exception profiles.
- Removing GA tiles and profiles.

Figure 4–1 shows the RDF Dashboard as seen in the UI. It consists of two Metric profiles and two Exception profiles.

Figure 4–1 RDF CS Dashboard



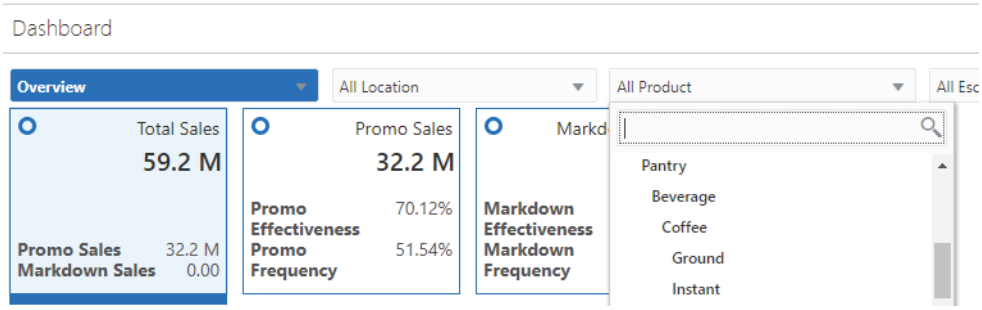
Note that the Exception profiles consist of Exception Tiles, and the Metric Profile consists of metric tiles of the type Comparison Tile. Currently RDF does not support the Variance Metric tile.

In Figure 4–1,the Overview Metric profile is selected and the Total Sales tile is highlighted with two sub-measures: Promo Sales and Markdown Sales.

Dashboard Intersection

The RDF GA Dashboard workbook is built at the Sub-class, District level which is controlled by the Dashboard Intersection specified in the Common plug-in. Refer to "Set Up Common Configuration Details." The Dashboard intersection also defines the level to which we can drill down the Product and Location filters in the Dashboard.

Figure 4–2 Product / Location Filters in the Dashboard



Process to Customize the Dashboard

Dashboard profiles correspond to a worksheet in the Dashboard workbook template in the configuration; and the measures displayed in the tiles are measures present in the worksheet corresponding to that profile. So customizing the dashboard is a three-step process:

1. In the Configuration, add the worksheet, measures, and rules to the Dashboard workbook template.
2. Regenerate the configuration by running the plug-in automation and then validate the configuration by running the rdf_config_validation script. Refer to the section, "Validating the Customized Configuration," for more information.
3. Customize the GA Dashboard Settings file in the Deployment Tool.

Note that the Deployment Tool is a utility within the Configuration Tools. Refer to the section, *Deployment Tool – Dashboard Settings Resource in the Oracle Retail Predictive Application Server Cloud Edition (RPASCE) Configuration Tools User Guide*.

The RDF GA Dashboard Settings configuration file is located in the configuration: **`RDF\plugins\RDFDashboardSettings.json`**

Steps to add a custom profile:

1. In the Configuration Tool, add custom worksheet and measures to the worksheet in the dashboard workbook template in the configuration. Also add load/calc rule for the measures.
2. In the Deployment Tool, open the GA Dashboard Settings configuration file.

3. Add the custom profile (Exception or Metric) to the Dashboard Settings configuration file.
4. Save the file in the Deployment Tool.

Steps to add a custom tile:

1. Identify the profile and worksheet to which the custom tiles need to be added.
2. In the Configuration Tool, add the custom measures to the corresponding worksheet. Also add load/calc rule for the measures.
3. In the Deployment Tool, open the GA Dashboard Settings configuration file.
4. Based on whether Exception or Metric profile, add the Exception tile or Comparison Metric Tile.
5. Save the file in the Deployment Tool.

Steps to remove GA tiles and profiles:

> **Note:** Do not remove the GA measures or worksheet from the Dashboard workbook template in the configuration.

1. In the Deployment Tool, open the GA Dashboard Settings configuration file.
2. Delete the GA profile or tile.
3. Save the file in the Deployment Tool.

Save the Dashboard Settings Configuration file in the same location in the configuration, that is:**`RDF\plugins\RDFDashboardSettings.json`**. Since this file is stored inside the configuration, whenever the customer uploads the configuration to the **`INCOMING_FTP_PATH`**, the customized Dashboard Configuration file will be used by the application during the domain build or patch process.

Once the domain is built or patched, if minor changes need to be done to the Dashboard that do not require a configuration change, then RPASCE provides a mechanism to Upload and Retrieve JSON files from the application.

This is supported through the **Configured Batch OAT task -> Manage JSON File**s. Refer to the *Oracle Retail Predictive Application Server Cloud Edition (RPASCE) Administration Guide* for detailed information on the OAT tasks.

Steps to Retrieve/Upload the Dashboard Configuration file:

1. Go to the **Configured Batch OAT task -> Manage JSON Files -> Retrieve option**.
2. The dashboard settings file will be downloaded into the **`OUTGOING_FTP_PATH`** as **`RDF_json.tar.gz`**
3. Un-tar the file and open it in the Deployment Tools.
4. Edit the file. Note that only minor updates that do not require a configuration change can be made at this time.
5. Save the file and zip it up as **`RDF_json.tar.gz`** and then upload it to the **`INCOMING_FTP_PATH`**
6. Then go to the **Configured Batch OAT task -> Manage JSON Files -> Upload option**.
7. Log out and log in to the client.

8. The Dashboard should be updated with the changes

Applying Changes to the Cloud Environment

To implement these changes in the cloud environment, it is necessary to either build a new domain or patch the domain. Refer to the *Install/Patch Domain* chapter in the *Oracle Retail Demand Forecasting Cloud Service Administration Guide*.

A

Appendix: Rules to Populate Out-of-stock and Outlier Indicator Measures

This appendix describes how to create rules to populate an Out-of-stock (OOS) and Outlier Indicator measures.

Rules Overview

In most RDF CS implementations, the out-of-stock and outlier flags are interfaced into RDF. However, not all retailers keep track of outages and outliers, or they are not very exact.

The following are some rules that populate these indicators. Note that they are a point of view, and you are encouraged to further refine them to fit your business needs.

OOS Rules Logic

If we call OOS the out-of-stock indicator and outliers the outlier flag, the logic to populate the flags can be:

If the rate of sales of an item is below a threshold, OOS is false.

Otherwise, make sure the item has been selling for a while and also does not have extensive periods with zero sales. If it is a new item, I do not want to start correcting the demand until the patterns become stable/predictable. Also, if the sales history has many zeroes, this may be an indication of a bigger issue, and the user may want to correct/take action in the Source Measure Maintenance workspace.

If these conditions are fulfilled, and the sales are still considered low, then mark the week as out-of-stock.

OOS Pseudocode

The pseudocode can look like:

```
If threshold 1a < rate of sales < threshold 1b
If minimum number of sales periods > threshold 2a
                        &&
      number of periods with zero sales < threshold 3a
                         &&
      sales < percent 1a * rate of sales
   then OOS = TRUE
 else if rate of sales > threshold 1b
   If minimum number of sales periods > threshold 2b
                         &&
      number of periods with zero sales < threshold 3b
                         &&
      sales < percent 1b * rate of sales
   then OOS = TRUE
```

Outlier Flag Rules Logic

For weeks with regular demand (no event, discount, and so on, is active), if the sales are deemed to be too high then mark the period as an outlier.

Outlier Flag Pseudocode

The pseudocode can look like:

```
If sales > rate of sales * multiplier
                         &&
   Period is not promoted
Then outliers = TRUE
```

B

Appendix: RDF Cloud Service integration with RMF Cloud Service

This appendix describes RDF Cloud Service integration with RMF Cloud Service

Integration Assumptions

Following is a list of requirements/assumptions for integration with RMF CS:

- RMF CS sends hierarchy and data files on a weekly basis. Hierarchy files are sent as full set. The data files are sent incrementally.
- RMF CS sends the data at item level.

Integrated Hierarchy and Data Files

RMF CS sends the following sets of hierarchy and data files. All RMF CS files are copied to the common cloud service share location from where RDF CS can pick up the files for further processing. Interface files need to be transformed to format the files into the RDF CS required format to filter columns not used and split and create different group of files as needed by RDF CS. RDF CS Configured Batch Process will transform the files to the RDF CS required formats, rename the final files as needed by RDF CS, and copy them to the domain input location for subsequent hierarchy or measure load processing. For each of the hierarchy and data files, RMF CS also sends a trigger file with the same name as the original file with the extension .complete~cloud. The RDF Cloud Service batch process waits for the trigger files to start processing the corresponding data or hierarchy file. After processing, it deletes those trigger files.

Hierarchy Files

The following hierarchy files are provided by RMF CS and need to be transformed to the RDF CS format before loading them into RDF Cloud Service. For each hierarchy file, RMF CS only sends the base and alternate dimensions of the hierarchies and not the user alternate rollups.

Calendar Hierarchy - rms_clnd.csv.dat

RMF CS sends the calendar hierarchy file without calendar labels. Calendar labels are added in the weekly batch process by calling the RDF CS custom script (ra_custom) with the parameter rms_rdf_clnd_label, which adds the calendar labels.

RMF CS sends the calendar hierarchy file with dates for five years. Based on the current date, two year historical dates, current year dates, and two future year dates.

Table B–1 contains the list of column contents from the file. Only required columns for RDF CS are filtered by the transform process.

Table B–1 Calendar Hierarchy File Fields

Field Name	Field Description	RDF CS Mapping
Day	The date from which the 4-5-4 data was derived, in YYYYMMDD format.	Day
Week	The end of week date for the day, in YYYYMMDD format.	Week
Month	The 4-5-4 month of the year, valid values 1 to 12.	Mnth
Quarter	The 4-5-4 quarter of the year, valid values 1 to 4.	Qtr
Half	The 4-5-4 half of the year, valid values 1 or 2.	Half
Year	The 4-5-4 year.	Year
week_of_year	The 4-5-4 week of the year, valid values 1 to 53.	WOYR
day_of_week	The day number within the week, valid values 1 to 7.	DOW

Note: When the RMS calendar extract is imported into RDF, the RMS month id 1 corresponds to February in RDF. It follows the 4-5-4 Retail Calendar. RDF has provided a numeric domain property ***app_start_month*** to override the default behavior. This can be done using OAT.

The default value of app_start_month =2, means RMS month id 1 will correspond to February.

If app_start_month = 10, means RMS month id 1 will correspond to October.

If app_start_month = 10, means RMS month id 12 will correspond to September.

Product Hierarchy - rms_prod.csv.dat

RMF CS exports all sellable and inventoried items.

Table B–2 contains the list of column contents from the file. Only required columns for RDF CS are filtered by the transform process. Some measure data files are also extracted and loaded from this RMF CS product hierarchy file.

Table B–2 Product Hierarchy File Fields

Field Name	Functional Name	RDF CS Mapping
ITEM	Item ID	sku
ITEM_DESC	Item Description	sku_label
ITEM_PARENT_DIFF	Parent/Diff ID	skup
ITEM_PARENT_DIFF_DESC	Parent/Diff Description	skup_label
ITEM_PARENT	Parent ID	skug
ITEM_PARENT_DESC	Parent Description	skug_label
SUBCLASS_ID	Subclass ID	scls

Table B–2 (Cont.) Product Hierarchy File Fields

Field Name	Functional Name	RDF CS Mapping
SUB_NAME	Subclass Name	scls_label
CLASS_ID	Class ID	clss
CLASS_NAME	Class Name	clss_label
DEPT	Department	dept
DEPT_NAME	Department Name	dept_label
GROUP_NO	Group	pgrp
GROUP_NAME	Group Name	pgrp_label
DIVISION	Division	dvsn
DIV_NAME	Division Name	dvsn_label
COMPANY	Company	cmpp
CO_NAME	Company Name	cmpp_label
FORECAST_IND	Forecastable Item flag	Not Applicable
CLASS_DISPLAY_ID	Class Display ID	Not Applicable
SUBCLASS_DISPLAY_ID	Subclass Display ID	Not Applicable
BRAND_NAME	Brand ID	Not Applicable
BRAND_DESCRIPTION	Brand Description	Not Applicable
SUPPLIER	Supplier Site	vndr
SUP_NAME	Supplier Name	vndr_label
DIFF_TYPE1	Diff Type1	Not Applicable
DIFF_ID1	Diff 1	Not Applicable
DIFF_TYPE2	Diff Type2	Not Applicable
DIFF_ID2	Diff 2	Not Applicable
DIFF_TYPE3	Diff Type3	Not Applicable
DIFF_ID3	Diff 3	Not Applicable
DIFF_TYPE4	Diff Type4	Not Applicable
DIFF_ID4	Diff 4	Not Applicable

Location Hierarchy - rms_loc.csv.dat

Table B–3 contains the list of column contents from the file. Only required columns for A&IP FSL CS are filtered by the transform process.

Table B–3 Location Hierarchy File Fields

Field Name	Functional Name	A&IP CS Mapping
LOCATION	Location ID	stor, strc
LOC_NAME	Location Name	stor_label,strc_label
DISTRICT	District ID	dstr
DISTRICT_NAME	District Name	dstr_label
REGION	Region ID	regn, tdar

Table B–3 (Cont.) Location Hierarchy File Fields

Field Name	Functional Name	A&IP CS Mapping
REGION_NAME	Region Name	regn_label, tdar_label
AREA	Area ID	chnl, chnc
AREA_NAME	Area Name	chnl_label, chnc_label
CHAIN	Chain ID	chan
CHAIN_NAME	Chain Name	chan_label
COMPANY	Company ID	comp
CO_NAME	Company Name	comp_label
COMPANY_CURRENCY	Primary Currency	Not Applicable
LOC_TYPE	Location Type	Not Applicable
LOC_TYPE_NAME	Location Type Description	Not Applicable
PHYSICAL_WH	Physical WH ID	Not Applicable
PHYSICAL_WH_NAME	Physical WH Name	Not Applicable
CHANNEL_ID	WH Channel ID	Not Applicable
CHANNEL_NAME	WH Channel Name	Not Applicable
STORE_CLASS	Store Class	stcl
STORE_CLASS_ DESCRIPTION	Store Class Description	stcl_label
STORE_FORMAT	Store Format	sfmt
STORE_FORMAT_NAME	Store Format Names	fmt_label

Product Attribute Hierarchy - rms_uda.csv.dat, rms_brand.csv.dat, rms_ vendor.csv.dat, rms_diff.csv.dat

UDA, Suppliers (Vendor), Brand, and Diffs in RMF CS are treated as product attributes in RDF Cloud Service. RMF CS provides them as individual files in the same format; those will be merged and loaded as one product attribute file. UDA_VALUE (Product Attribute Values) are not unique across UDA_ID in RMF CS. While transforming and loading as a product attribute, UDA_ID position will be concatenated with UDA_ VALUE using an underscore ("_").

Table B–4 contains the list of column contents from the file. Only required columns for A&IP FSL CS are filtered by the transform process.

Table B–4 Product Attribute Hierarchy File Fields

Field Name	Functional Name	RDF CS Mapping
UDA_ID	UDA	Patt
UDA_DESC	UDA Description	patt_label
UDA_VALUE	UDA Value	patv
UDA_VALUE_DESC	UDA Value Description	patv_label

RDF Cloud Service Output Data

The following data files are provided by RMF CS and need to be transformed to the RDF CS format before loading them into RDF Cloud Service. All RMF CS files are

transformed first and loaded together using the single Load Task Load - RMS Interface Data.

Sales Data File - wsales.csv.ovr

Table B–5 contains the list of column contents from the file. Only required columns for RDF CS are filtered by the transform process. RMF CS sends Net Sales in all regular, promotion, and clearance buckets. The transform process will split the sales into three files, rsal.csv.ovr, psal.csv.ovr and csal.csv.ovr based on the buckets.

Note: The Other Sales (osal) measure has not been currently integrated with RMFCS and can only be loaded into RDFCS using a flat file.

Table B–5 Sales Data File Fields

Field Name	Functional Name	RDF CS Mapping
ITEM	Item ID	sku
LOCATION	Location ID	stor
EOW_DATE	End of Week Date	week
SALES_UNITS	Sales Unit	rsal,psal or csal
SALES_TYPE	Sales Type (regular, promotion or clearance)	Used to divided sales units into different files

Outage Indicator Data File - ldpreoosind.csv.ovr

Table B–6 contains the list of column contents from the file. Only required columns for RDF CS are filtered by the transform process.

Table B–6 Outage Indicator Data File Fields

Field Name	Functional Name	RDF CS Mapping
ITEM	Item ID	sku
LOCATION	Location ID	stor
EOW_DATE	End of Week Date	week
INDICATOR	Outage indicator	ldpreoosind

Item Attribute Mapping File - prdattt.csv.ovr

Table B–7 contains the list of column contents from the file. Only required columns for RDF CS are filtered by the transform process. This file only contains Product Attribute mappings for UDA attributes. Item Attribute mappings for Brand, Supplier, and Diffs are derived and loaded from the RMF CS Product file.

Table B–7 Item Attribute Mapping File Fields

Field Name	Functional Name	A&IP CS Mapping
ITEM	Item ID	sku
UDA_ID	UDA	patt
UDA_DESC	UDA Description	Not Applicable
UDA_VALUE	UDA Value	prodattt

Table B–7 (Cont.) Item Attribute Mapping File Fields

Field Name	Functional Name	A&IP CS Mapping
UDA_VALUE_DESC	UDA Value Description	Not Applicable
FORECAST_IND	Forecastable Item	Only items that have the flag set to True are imported into RDF CS.

Integrations Between RDF Cloud Service and other Applications

This section details:

- Batch Tasks Specific to RMF CS Integration
- Implementation Steps for RMF CS Integration

Batch Tasks Specific to RMF CS Integration

The following sets of Online Administration Tools tasks are pre-configured in Configured Batch Tasks to support RMF CS integration. For more details about running the tasks, see the *Oracle Retail Retail Demand Forecasting Administration Guide*.

- Configured Batch Task > Run Batch Measure Export Group > Export Daily Demand

  This task exports daily forecast and cumulative interval into a file named dsdemand.csv.ovr and push it to SFTP output and the common cloud service share location.

- Configured Batch Task > Run Batch Measure Export Group > Export Weekly Demand

  This task exports weekly forecast and cumulative interval into a file named wsdemand.csv.ovr and push it to SFTP output and the common cloud service share location.

- RDF Cloud Service Batch Task > Run RDF Cloud Service Batch > Export Demand

  This task performs the work of both Export Daily Demand and Export Weekly Demand.

- RDF Cloud Service Batch Task > Run RDF Cloud Service Batch > Transform RMS File

  This task only transforms all RMS hierarchy and data files to the RDF CS required format and copies the hierarchy files to the SFTP output as rms_hier.zip. It will not load those files.

  Typically, this can be used during initial installation to transform the initial set of RMF CS files and rebuild the domain with the initial set of RMF CS hierarchy files.

Implementation Steps for RMF CS Integration

If RMF CS needs to be integrated as part of an initial implementation, then the following needs to be done in this order:

1. Install RDF Cloud Service using the Installer which will install the default solution with GA data files.
2. Upload the RMF CS Hierarchy and data files for the initial conversion data.

3. Run the Transform RMS Files batch task to transform hierarchy files to RDF CS format. Those will be copied to SFTP output.
4. Upload the generated hierarchy files and also additional hierarchy files needed for RDF CS, that are not provided by RMF CS (use the GA data set as a reference to create additional hierarchy files) into the SFTP server, at the INCOMING_FTP_PATH/input subdirectory.
5. Using the Self-Service Administration Task in the bootstrap domain, rebuild the domain with the required plug-in options for RDF CS and without selecting Use GA Data. The process will rebuild the domain using the initial set of RMF CS hierarchy files.
6. After building the domain, re-send all the hierarchy and data files from RMF CS and use the regular Weekly Batch to process the hierarchy and data files.

C

Appendix: RDF Cloud Service Integration with PDS and BDI

This appendix describes how RDF Cloud Service supports integration with RMF CS using Bulk Data Interface (BDI) and Planning Data Storage (PDS). It also allows using only PDS to share data across different RPAS Planning applications if they are registered against it and have common shared hierarchies and positions.

For information on the BDI process flow, see the BDI Process Flow appendix in the *Oracle Retail Predictive Application Server Cloud Edition Implementation Guide*.

Using PDS

RPAS applications can share measures across shared applications if they share common hierarchies and measure data. RPAS has a PDS which can store the shared dimensions and fact data (shared measures).

A PDS-enabled domain needs an Integration Configuration which contains Shared Dimension data and also shared measures as Fact data that is grouped as Fact Group across different RPAS applications.

All RPAS applications that can be integrated using PDS share a common GA Integration Configuration which is part of the RPAS Application Standard Libraries (RASL) package called RGBU_RDM.xml that is pre-configured.

The following section describes the shared dimension and measure data from the RDF CS solution which is present in Integration Configuration.

Using BDI

BDI allows sharing data across the enterprise. RPAS have BDI importers that import the data from the RMF CS database tables into the PDS. If the customer implements RMF CS and BDI, then the use of PDS is mandatory, as currently the data loaded into BDI can only be internally transformed to PDS shared measures. For more details about PDS, see Using PDS.

If BDI is enabled, there is no need for retailers to export flat files from RMS. In order for the batch process to not wait for the files from RMS, the Batch Uses BDI Boolean flag needs to be set to True.

Shared Hierarchies

Shared Hierarchies in PDS should have the union of all dimensions across the different RPAS applications without any User Defined Dimensions and also all of them should

have the same names for common hierarchies and common shared dimensions. If RMF CS integration and BDI are used, those hierarchies will be loaded as part of that scheduled process in BDI.

If the customer is only using PDS and not using BDI, the customer has to first load the shared hierarchies into one of the RPAS applications by calling the loadDimData utility by using Standard Admin Tasks after uploading the shared dimension file named as <hier>.csv.dat with header information including all shared dimension positions with position labels. In the header, the dimension name should be the shared dimension name in the hierarchy file and the dimension label should be <dim>_label.

The file should be uploaded into the $INCOMING_FTP_PATH/rdm_input/dimdata directory.

If the domain is registered against PDS, then the Batch Uses PDS Boolean flag will be set during batch. It controls the flow of the weekly batch.

The RDF CS weekly batch, in turn calls the batch framework service dimdataload to load the dimension data into PDS, if the domain is registered against PDS (Batch Uses PDS flag is set) and not using BDI (Batch Uses BDI flag is not set) to get its data.

Loading dimension data will only load the dimension details into the PDS. The customer needs to run the standard loadHier utility without any input files, in order to synchronize the domain dimension data with PDS for updated positions. That will also be run by default as part of the weekly batch process.

The following tables, Table C–1 through Table C–4, list the shared hierarchies and dimensions for RDF CS available in the GA Integration Configuration. Any additional dimensions present in these hierarchies which are not present in RDF CS can be from another shared application. Those also need to be provided if the customer loads those positions into PDS.

Table C–1 Calendar Hierarchy (clnd)

Shared Dimension	Shared Dimension Label	Aggregate Dimension
day	Day	Not Applicable
week	Week	day
mnth	Month	week
qrtr	Quarter	mnth
half	Half	qrtr
year	Year	half
woyr	Week of Year	year
dow	Day of Week	day

Table C–2 Product Hierarchy (prod)

Shared Dimension	Shared Dimension Label	Aggregate Dimension
sku	Item	Not Applicable
skup	Style/Color	sku
skug	Style	skup
scls	Subclass	skug
clss	Class	scls
dept	Department	clss

Table C–2 (Cont.) Product Hierarchy (prod)

Shared Dimension	Shared Dimension Label	Aggregate Dimension
pgrp	Group	dept
dvsn	Division	pgrp
cmpp	Company	dvsn
brnd	Brand	sku
vndr	Vendor	sku

Table C–3 Location Hierarchy (loc)

Shared Dimension	Shared Dimension Label	Aggregate Dimension
stor	Location	Not Applicable
dstr	District	stor
regn	Region	dstr
chnl	Channel	regn
chan	Chain	chnl
comp	Company	chan
loct	Location Type	stor
phwh	Physical Warehouse	stor
fflt	Fulfillment Type	stor
tdar	Trade Area	stor
strc	Store Cluster	stor
chnc	Cluster Channel	strc
sfmt	Store Format	stor
stcl	Store Class	stor

Table C–4 Product Attribute Hierarchy (patr)

Shared Dimension	Shared Dimension Label	Aggregate Dimension
patv	Product Attribute Value	Not Applicable
patt	Product Attribute Type	patv

Note: If the customer implementation uses RMF CS flat files for integration and are not using BDI but would like to also use PDS, then it is only partially supported, customer has to customize the batch transform process to transform the RMS files to create the shared hierarchy files from the RMS files as needed by PDS with headers.

Note: GA batch control files contain entries only to load shared hierarchies used by the application into PDS, but PDS can have more hierarchies not used by the current application. If not using BDI to load PDS, it is recommended to load the shared hierarchy common across multiple applications into only one application which contains most of the shared hierarchies. That way the customer can avoid uploading shared hierarchies data into multiple applications.

Shared Fact Data

PDS shared measure data are called as Fact Data to share across different planning applications.

Shared fact data are grouped together as fact groups to enable them to be accessed together. If RMF CS Integration and BDI are used, those shared fact measures are directly loaded as part of the scheduled process in BDI. If the customer is only using PDS and not using BDI, then they have to load the fact data into one of the RPAS applications by calling the loadFactData utility by using Standard Admin Tasks after uploading the shared fact file named as <fact_group>.csv.ovr or <fact_group>.csv.rpl with the header as the shared fact name. The customer can use any single file with any name or split them into multiple files to load different fact measures if all of them have the same base intersection, but it is recommended to load the files after grouping them as per the fact groups to allow them to load in parallel.

The file should be uploaded into the $INCOMING_FTP_PATH/rdm_input/factdata directory.

If the domain is registered against PDS, then the Batch Uses PDS Boolean flag will be set during batch. It controls the flow of weekly batch.

Loading fact data will only load the fact data into PDS and will automatically make the mapped shared measures available into the registered domains, customer does not have to load those shared measures again into domains.

Table C–5 lists the shared measures for RDF CS available in the GA Integration Configuration.

Table C–5 RDF CS Shared Measures Available in the GA Integration Configuration

Fact Name	Fact Label	Fact Group Name	Shared Measure Name
psal	Promotion Sales	psal	psal
rsal	Regular Sales	rsal	rsal
csal	Clearance Sales	csal	csal
LdPreOosInd	Loaded Out of Stock Indicator	wsal	PreOosInd
appwkfrcst	Approved Forecast	fcst	appwkfrcst
appwkbaseline	Approved Baseline	Fcst	appwkbaseline
appwkcumint	Approved Forecast Error	Fcst	appwkcumint
appdayfrcst	Approved Forecast - Day Level	fcst_day	appdatfrcst
appdaybaseline	Approved Baseline - Day Level	fcst_day	appdaybaseline

Table C–5 (Cont.) RDF CS Shared Measures Available in the GA Integration

Fact Name	Fact Label	Fact Group Name	Shared Measure Name
appdaycumint	Approved Forecast Error - Day Level	fcst_day	appdaycumint

Note: If you want to integrate RDF CS with RMF CS using flat files and use PDS but not use BDI, then it is only partially supported. You have to customize the batch transform process to transform the RMS files to fact files with headers fact names as headers to load into PDS.

Note: Currently BDI-PDS existing integration is not extensible, it cannot be further customized to add more measures to be integrated from BDI. Only existing measures present in GA Integration configuration for PDS can now be integrated with BDI.

Note: PDS allows using fact measures with a lower intersection than the shared measures present in the domain and allows them to aggregate it while accessing the data from PDS. For MFP CS, all fact measures loaded at the sku level are aggregated and viewed as shared measures in MFP CS at the scls level.

Implementation Steps with BDI, PDS, and RMF CS Integration

If BDI implementation with RMF CS is used for implementation, the following steps need to be done in this order. The steps assume that RPAS, RASL, UI, and BDI are already deployed:

1. Once MFP Cloud Service environment is provisioned, using bootstrap build MFP domain using the initial set of customer hierarchy data. For Shared Hierarchies which are expected to come from BDI/PDS, load only with valid positions which are in BDI. It's not necessary to have all records, but atleast one valid position in each of the shared hierarchy files is needed.
2. Build PDS using the default integration configuration, without uploading any data. After PDS is built, it will also register the domain against that PDS.
3. Integrate with BDI and load PDS by running the BDI Process Flows. That should load the PDS schemas with data from BDI for both shared dimensions and shared measure (fact) data.
4. Upload the hierarchy files for non-shared hierarchies and run the Load Hierarchies OAT task under Configured Batch Tasks. It will synch up the shared hierarchies with PDS and load the non-shared hierarchies from the uploaded files.
5. Set the Batch Uses BDI and Enable RMF CS Boolean measures in the Admin workbooks.
6. Set up the necessary Admin data or load the admin data files.
7. Run the weekly batch to process all loaded actuals (from PDS for shared data).

D

Appendix: New Item Configuration Guide

This appendix describes how RDF Cloud Service performs the batch process for New Items and alternative implementation approaches for the New Item module.

New Item Batch Tasks

New Item batch performs the following tasks:

1. Identify the current New Items that will turn into existing items. This identification is based on the approved Like Item, approve substitution method, sales history start date and time series duration. If a New Item is going to turn into an existing item, its approved Like Items, substitution method and settings will be automatically wiped out.
2. Automatically identify New Item/stores in the system based on the user loaded New Item forecast start date
3. Calculate eligible Like Items for the auto-identified New Item/store. The eligible criteria is based on the previous approved base demand, existing items's sales history length, and if the existing item is under the same user specified level as the New Item. The Like Item eligibility criteria also includes a Like item exclusion mask that can be customized based on business needs. The user specified level is an input in New Item plug-in.
4. Calculates the similarity score between the New Item and eligible Like Items based on product attribute information and provides three Like Item recommendations per the New Item/store that is based on the similarity score rankings.
5. Automatically approves system recommended Like Items, substitution methods, and parameters if user specified.

For details about New Item substitution methods and settings, refer to the *Oracle Retail Demand Forecasting Cloud Service User Guide*.

New Item Forecast Approaches

In RDF Cloud Service, New Item forecasting is implemented differently from previous versions. A baseline forecast for the New Item is generated based on the seasonal curve level. The seasonal curve for the New Item is produced through seasonality escalation like other existing items. The level for the New Item can be produced based on settings in new item module.

Implementation Options

In a normal RDF Cloud Service implementation, there are four possible approaches to setting up the New Item module.

First Approach

Ignore

The customer does not plan to use it and chooses to simply ignore the New Item. For this approach New Items need to be set up for the RDF plug-ins to run successfully. Do the following to ignore using the New Item:

1. Leave all the GA New Item plug-in inputs unchanged.
2. Use the GA attribute hierarchy file for the domain build.
3. After the domain build, open the Batch Flow Management workbook and set up the boolean switches related to the New Item as follows:
   a. Set **`runnewitembatch`** to false
   b. Set **`runnewitem_BF_`** to false
   c. Set **`runnewitem_CF_`** to false
   d. Commit the workbook changes

Second Approach

Use New Item to Manually Set Up New Item Forecasting Parameters

The customer does not have any attribute information, but would like to use New Item to manually set up New Item Forecast Parameters such as Like Item, Like Store, Substitute Method, user provided Base Demand, and so on. The New Item batch is still necessary to run in this case, because the first step of New Item batch is necessary to automatically turn off New Item functionality once a New Item matures. However, the Attribute Maintenance workbook will not be able to be created because there is no valid attribute information.

For this approach, the implementor need to set up the New Item using the following steps:

1. Go over the following New Item plug-in inputs one by one:
   - New Item data source

     The intersection of the specified measure decides the level where the New Item and New Store will be set up.
   - Product Map

     Allows the user to specify which level that the New Item and eligible Like Item should be under. A Like Item can only be selected from the items that roll up to the same level.
   - New Store level

     Allows users to specify Like Stores for the whole level
2. Use the GA attribute hierarchy file for the domain build.
3. After the domain build, open the Batch Flow Management workbook and set up the boolean switches related to New Item as follows:

Note: New Item forecasting will generate unpredictable results if the final level intersection's prod and loc dimensions do not match what is configured in the New Item module.

a. Set **`runnewitembatch`** to true.

b. Set *runnewitem_BF_* to true if the level needs to enable New Item forecasting. The final level forecast intersection must have the same product and location level as the New Item data source measure.

c. Set *runnewitem_BF_* to false if the level needs to disable New Item forecasting or the final level intersection 's product and location level were different from the New Item data source measure.

d. Set *runnewitem_CF_* to true if the level needs to enable New Item forecasting. The final level forecast intersection must have the same product and location level as the New Item data source measure.

e. Set *runnewitem_CF_* to false if the level needs to disable New Item forecasting or the final level intersection 's product and location level were different from the New Item data source measure.

f. Commit the workbook changes.

4. Use New Item Maintenance workbook and the New Store Maintenance workbook to set up New Item and New Store settings in the New Item Basic Parameters worksheet of New Item Maintenance Workbook and make sure *nitautoapp* is set to false.

   This will turn off *autoapprove* in the New Item batch process. Set *nitdtsdur* (time series duraton) to the user desired value. This control how many sales periods a New Item needs to mature.

Third Approach

With Attribute Information, Use New Item to Get Recommended Like Items and Manually Set Up Like Item and Like Stores

The customer has attribute information and would like to use New Item to get recommended like items and manually set up like item and like stores. Full fledged New Item functionality is enabled in this case and data preparation is very important. In this case, the implementor needs to set up the New Item using the following steps:

1. Go over the following New Item plug-in inputs one by one:

   - New Item data source

     The intersection of the specified measure decides the level where the New Item and New Store will be set up.

   - Product Map

     Allows the user to specify which level that the New Item and eligible Like Item should be under. A Like Item can only be selected from the items that roll up to the same level.

   - Product Map RHS

     Right hand side of the Product Map.

   - New Store level

Allows users to specify Like Stores for the whole level

- Product RHS

  This should be consistent with the New Item data source.

- Location RHS

  This should be consistent with the New Item data source.

2. Prepare the following data files for domain build and ongoing data load in the future:

   - **`PATR.csv.dat`**—attribute hierarchy file
   - **`prdattT.csv.ovr`**—product attribute value data file
   - **`nitdattwgt.csv.ovr`**—attribute weight data file (without this data file, there is no similarity score and Like Item recommendation)
   - **`nitfcststovr.csv.ovr`**— New Item forecast start date data file (without this data file, there is no New Item identification)
   - **`nisros.csv.ovr`**—user provided New Item base rate of sales data file.

   Without proper data, the Attribute Maintenance workbook can not be created in the future.

3. After the domain build, open the Batch Flow Management workbook and set up the boolean switches related to New Item as follows:

   Note: New Item forecasting will generate unpredictable results if the final level intersection's prod and loc dimensions do not match what is configured in the New Item module.

   a. Set **`runnewitembatch`** to true

   b. Set **`runnewitem_BF_`** to true if the level needs to enable New Item forecasting. The final level forecast intersection must have the same product and location level as the New Item data source measure.

   c. Set **`runnewitem_BF_`** to false if the level needs to disable New Item forecasting or the final level intersection 's product and location level were different from the New Item data source measure

   d. Set **`runnewitem_CF_`** to true if the level needs to enable New Item forecasting. The final level forecast intersection must have the same product and location level as the New Item data source measure.

   e. Set **`runnewitem_CF_`** to false if the level needs to disable New Item forecasting or the final level intersection 's product and location level were different from the New Item data source measure

   f. Commit the workbook changes.

4. Use New Item Maintenance workbook and New Store Maintenance workbook to set up New Item and New Store settings in the New Item Basic Parameters worksheet of New Item Maintenance workbook, set up **`nitautoapp`** to the user desired value. This will enable or disable **`autoapprove`** in the New Item batch process. Set *nitdtsdur* (time series duraton) to the user desired value. This control how many sales periods a New Item needs to mature

5. If the loaded New Item forecast start date and New Item base rate of sales were not available, the following business flow must be performed:
   a. Open the New Item Maintenance workbook to input **nitfcststovr** and **nisros** without running any approve custom menu and then commit the change.
   b. Run **newitem_batch** to automatically identify New Items and generate system recommended like SKU.
   c. New Items can be approved automatically in Step b. The user can also create the New Item Maintenance workbook again to manually approve the New Item settings.

Fourth Approach

Bypass New Item Workbooks and Set Up New Item Forecasting through Direct Measure Loading and Calculation

Use this approach when you want to use New Item Forecast, but do not have any attribute information and you do not want to populate New Item parameters manually. You can load information into the New Item Parameter measures directly and avoid using any New Item workbook. It is still necessary to run the New Item batch because the first step of New Item batch automatically turns off New Item functionality once a New Item matures.

For this approach, the implementor needs to set up the New Item using the following steps:

1. Verify the following New Item plug-in inputs one by one:
   - **New Item data source**—the intersection of the specified measure decides the level where the New Item and New Store will be set up.
   - **Product Map**—allows the user to specify which level that the New Item and eligible Like Item should be under. A Like Item can only be selected from the items that roll up to the same level.
   - **New Store level**—allows users to specify Like Stores for the whole level
2. Build the domain using the GA attribute hierarchy file.
3. After the domain build, open the Batch Flow Management workbook and set up the boolean switches related to New Item as follows:

   Note: New Item Forecasting generates unpredictable results if the final level intersection's **prod** and **loc** dimensions do not match with what is configured in the New Item module.

   a. Set **runnewitembatch** to true.
   b. Set **runnewitem_BF_** to true if the level needs to enable New Item forecasting. The final level forecast intersection must have the same product and location level as the New Item data source measure.
   c. Set **runnewitem_BF** to false if the level needs to disable New Item forecasting or the final level intersection 's product and location level were different from the New Item data source measure.

d. Set **runnewitem_CF_** to true if the level needs to enable New Item forecasting. The final level forecast intersection must have the same product and location level as the New Item data source measure.

e. Set **runnewitem_CF_** to false if the level needs to **_disable** the New Item forecasting or the final level intersection 's product and location level were different from the New Item data source measure.

f. Commit the workbook changes

4. In the New Item Maintenance Workbook, ensure that **nitautoapp** is set to false to turn off auto approve in the New Item batch process. Set **nitdtsdur** (time series duration) to the user desired value. This controls how many sales periods a New Item needs to mature.

5. Prepare the custom batch control file that loads settings into measures such as **nisros** (user provided base demand), **nitdtsdur** (time series duration), **nitappsubm** (substitution method), **nitapplkitm1** (like item), **nitappadj** (like Item adjust percentage).

Note: In the weekly forecast batch, the New Item batch is run before the RDF batch forecast. The initial step in New Item batch identifies matured New Items and clears out its substitute method. RDF batch checks the substitution method to decide if an item goes through a normal forecasting process or New Item forecasting.

If an implementor wants to load a substitute method or other new item parameters, it is important to decide when it should be loaded, either before or after the New Item batch runs. If a substitute method is loaded before the New Item batch runs, it may be automatically wiped out depending on how long its sales history is. If a substitute method is loaded after the New Item batch, then the item will be subject to New Item forecasting forever.

Troubleshooting Tips for Common New Item Issues

The following tips will provide troubleshooting solutions to common issues for New Items in RDF CS.

Issue 1

Issue: The New Item Maintenance Attribute Maintenance Workbook Cannot be Opened Due to an Empty Pre-range Solution

Solution

If you are running into this issue you have by-passed Oracle's standard domain building or data loading process.

The New Item Maintenance workbook is pre-ranged by these measures:

- **flagllc**
- **nitwkrange**
- **nititm2clsr**

The **flagllc** measure ranges the prod hierarchy to llc items. The **nitwkrange** measure ranges the CLND hierarchy. The **nititm2clsr** measure ranges the pror hierarchy. The

nitwkrange measure is calculated in the new item batch, where as the **flagllc** and **nititm2clsr** measures are calculated as a post-process after the data load. The Calendar has been pre-ranged to bring in weeks > (TODAY - TS duration). The **flagllc** measure is calculated using **flagslc** when **flagslc** is set to the *False* default. Therefore, even if no data file is provided for **flagslc**, **flagllc** should be properly calculated. The **nititm2clsr** measure is calculated based on **nititm2itr**. The **nititm2itr** data file is generated by RDF data load process based on product hierarchy and loaded automatically. These logics were implemented through batch control file and OAT.

Users should not encounter any issue if the standard RDF process is followed.

If you encounter this issue after by-passing your domain, make sure that **nititm2itr**, **nititm2clsr**, and **llcflag** are populated in your domain.

Issue 2

Issue: The New Item Management Workbook Can Not be Opened Due to an Empty Prerange

Solution

Check to see if the measures **nititm2itr** and **nititm2clsr** are populated. The **nititm2itr** measure should be loaded automatically by RDF CS whenever product hierarchy is loaded.The **nititm2clsr** measure is recalculated based on **nititm2itr** after each dataload. These logics were implemented through OAT. Users should not encounter this issue if RDF Cloud Service's standard process were followed.

Another measure to check is **nitfcststovr** (forecast start date override). This measure is used to identify new item. It need to be a date in the future so that the item can be identified as new item.

Issue 3

Issue: Unable to Open the New Item Management Workbook Even if the nitfcststovr Measure is Populated and All the Other Measures are Properly Populated

Solution

The New Item batch will calculate the similarity score and pre-range mask for items that have not yet matured based on TS duration..

Another unlikely scenario is that no eligible Like Item can be identified for the New Item. The Similarity Score is only calculated between a New Item and an existing item. An existing item for a New Item must satisfy the following criteria:

- The existing item is under the same product category as the New Item. The level of product category is specified in New Item plug-in. In RDF GA configuration, it is class.
- The existing item must have attribute information.
- The existing item must have enough Sales History Length. The Sales History Length for a time series is calculated as the number of periods between today and the week that has the first populated sales. The Sales History Length must be more than the value specified in ts-duration (**nitdtsdur**). This situation is unlikely unless no sales data is available.

Check the eligible existing items by looking into the measure **nitlkitmlocmsk**. If it is empty, that means one of the condition is not meet. Check **prdAttT** for product attribute information. Check the sales input measure specified in the New Item plug-in for the Sales History Length.

Issue 4

Issue: You Can Not Generate a Forecast for a New Item Even If There are Valid Like Item and Substitute Method Settings

Solution

The New Item forecast is produced by looking up the seasonal curve through escalation and getting a base rate of sales from the like item/user input/class average escalation. After a forecast, you can check measures **woyseascurve_CF_** or **woyseascurve_BF_** to see if a seasonal curve has been generated for the New Item.

If it is not populated, that means there is no seasonal curve for all of the escalation intersections at this particular item/store. An implementer should consider adding a new catch-all escalation level. The other measures to check are **mergedlvl_BF_** or **mergedlvl_CF_**, if the measure is empty for the New Item/store that means that no base rate of sales is generated. You can further looking into the reason by base rate of sales cannot be generated.

Issue 5

Issue: The New Item Forecast is Setup in the New Item Maintenance Workbook, but No Forecast is Generated for the New Item and Measures mergedlvl_BF_ and mergedlvl_CF_ have a Zero Value

Solution

Check the **runnewitem_CF_** and **runnewitem_BF_** measures which are visible in the Batch Flow Management workbook. If the measure was false, set them back to true, commit and rerun forecast.

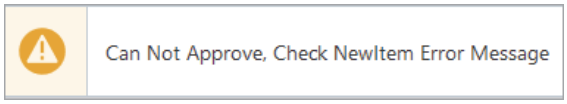
If **runnewitem_CF_** and **runnewitem_BF_** were true, then check the **nitappsubm** substitution method.

If nitappsubm is set to:	Then:
None	No New Item forecast will be generated.
Like Item	At least one valid existing item must be assigned as the Like Item. The Like Item itself must have a forecast. The Like Item assignment measure is: ■ **nitapplkitm1** The **nitappadj** (total adjustment percentage) can not be zero.
User Input	The user must provide the base rate of sales for the New Item. The value will be stored in **nitappros** after the New Item approval process. When the substitute method is user input, then **nitappros** can not be zero.

Issue 6

Issue: When Trying to Approve Like Item Recommendations, Received this Approve Error Message: *Can Not Approve, Check New Item Error Message*

Figure D–1 Error Message



Solution

This error is due to a problem in your New Item set up. On the worksheet, display show/hide for *select and approve.* For the New Item error message measure, ensure that measure is visible on the worksheet. It will provide additional clues for the issue.

E

Appendix: RDF Cloud Service Integration with ORASE / RI

This appendix describes how RDF Cloud Service supports integration with Oracle Retail Advanced Science Engine (ORASE) and Oracle Retail Insights (RI).

Integration with ORASE

RDF CS supports integration with ORASE to import the Demand Transference multiplier, import Size Profile Data, and export the forecast and promotion effects.

Import DT Multiplier

RDF CS can import the Demand Transference (DT) Multiplier from ORASE and use it to calculate the DT Effects if DT is enabled.

ORASE exports the DT multiplier file **`dtmul.csv.ovr`** to the Cloud Share location compressed as the **`ORASE_WEEKLY_ extract.zip`** file, together with the trigger file with the extension, **`.complete`**.

The customer can schedule the Import RI/DT Measures OAT task to import the DT multiplier. This process also loads the **`dtmul`** measure.

Import Size Profile

RDF CS can import the size profile generated by Size Profile Optimization (SPO) to spread the short lifecycle forecast from **skup** to **sku** level.

SPO exports the file **`spo_size_profile.csv`** to the Cloud Share location compressed as the **`ORASE_WEEKLY_ extract.zip`** file.

The customer can schedule the Online Administration Task Import Size Profile which extracts the required columns from the input file for the rows set as Y for column USED_BY_RDF. This process creates the file **`sizeprofile.csv.ovr`** at **sku/stor** level and load the **`sizeprofile`** measure.

Export Demand to RI

RDF CS exports the forecast and promotion related data to RI. RI serves as a central port for the retailer for all the insights. Table E–1 describes the files exported by the Export RI Base Demand OAT task. Each file exported to RI has a corresponding control (**`.ctx`**) file.

Table E–1 Files Exported by the Export RI Base Demand OAT Task

File #	Measure	Intersection	File Name	Control File
1	Baseline System Forecast	sku/store/week	W_RTL_PLANFC_PROD1_LC1_T1_FS.dat	W_RTL_PLANFC_PROD1_LC1_T1_FS.dat.ctx
	Baseline Approved Forecast			
	Baseline Approved Cumulative Intervals			
2	System Forecast	sku/store/week	W_RTL_PLANFC_PROD2_LC2_T2_FS.dat	W_RTL_PLANFC_PROD2_LC2_T2_FS.dat.ctx
	Approved Forecast			
	Approved Forecast Cumulative Intervals			
3	LLC Causal Variables	lprm	W_RTL_PLANFC_PARAM_DS.dat	W_RTL_PLANFC_PARAM_DS.dat.ctx
	SLC Causal Variables	sprm		
4	LLC causal effects	sku/stor/lprm	W_RTL_PLANFC_PARAM_IT_LC_DS.dat	W_RTL_PLANFC_PARAM_IT_LC_DS.dat.ctx
	SLC causal effects	sku/stor/sprm		
5	Baseline Approved Forecast	sku/store/week	W_RTL_PLANFC1_PROD1_LC1_T1_FS.dat	

Sample Entries for the Exported Files

W_RTL_PLANFC_PROD1_LC1_T1_FS.dat

```
2019-04-04;00:00:00|sku123456|stor104|1234.1234|1234.1234|22.33
```

W_RTL_PLANFC_PROD2_LC2_T2_FS.dat

```
2019-04-04;00:00:00|sku123456|stor104|1234.1234|1234.1234|22.33
```

W_RTL_PLANFC_PARAM_DS.dat

```
pvarlpr1 |boolean|linear
pvarlpr2|boolean|linear
pvarlpr3|real|exponential
pvarspr2|boolean|linear
pvarspr3|real|power
```

W_RTL_PLANFC_PARAM_IT_LC_DS.dat

```
sku123456|stor104|pvarlpr1|2.1
sku123456|stor104|pvarlpr2|2.5
sku123456|stor105|pvarlpr3|1.5
sku123456|stor106|pvarspr2|2.5
sku123456|stor107|pvarspr3|1.5
```

F

Appendix: Configuring Batch Run

This appendix describes how to configure a batch run for RDF Cloud Service. Currently RDF CS has a Daily batch configured to export forecast. There has been inquiry about additional functionalities to be included in daily batch such as applying promotion effects during the week and producing forecast for new items that were set up during the week.

Requirements

The following requirements can be satisfied by configuring the weekly batch to run daily. However there are several issues that an implementor must pay attention before letting **`batch_weekly`** run daily.

- The actual sales can be updated daily or weekly depends on set up. The current week's sale history will only zero or partial week. The current week's sales history should not be used in any process. When running daily batch every day, make sure your forecast start date is same as your last weekly batch forecast.
- Pre-processing does not need to be re-run for daily and it can be turned off using the **`runpreprocess`** measure
- The New Item batch can be optional depending on your business requirements. If there is no need to provide forecast for newly identified item during the week, it can be turned off.
- Users may have started manually adjusting the forecast in the forecast review workbook since the last weekly batch run, it is up to the implementor to decide if they want the daily batch result to override the user's adjustment. It will be a complex issue for the implementor if the user adjustment need to be preserved.

Configuration

The following process shows how to use extensibility to configure the weekly batch to run daily.

1. Perform the following activities for Step 1.
   a. Create a custom scalar boolean measure **`c_batchDailyB`** to distinguish Daily from the Weekly Batch Run.
   b. Configure a custom rule to calculate the **`c_batchDailyB`**. The expression can be written as

```
c_batchDailyB= position([clnd].[dow] ,index([clnd].[dow],
now))!="SUN"
```

If today is Sunday, **c_batchDailyB** is false, else **c_batchDailyB** is true.

When **c_batchDailyB** is true, daily batch should be invoked, else weekly batch is invoked.

This rule should be store a custom rule group that is invoked in the **hook_pre_load**.

2. Perform the following activity only when **c_batchDailyB** is true.

   a. Create custom measures and rule group to back up the following forecast parameter values:

```
deffrcststartdt07 : c_deffrcststartdt07=deffrcststartdt07
 defadjfrcstmth07 : c_defadjfrcstmth07=defadjfrcstmth07
 adjfrcstmthint07 : c_adjfrcstmthint07=adjfrcstmthint07
 adjfrcstmthovr07 : c_adjfrcstmthovr07=adjfrcstmthovr07
 runpreprocess : c_runpreprocess = runpreprocess
 runnewitembatch: c_runnewitembatch = runnewitembatch
```

   The previous expressions should be put in a **rulegroup** that is run in the **hook_pre_preprocess**

   b. Set the previous parameters to implementer desired values, such as:

```
deffrcststartdt07 : deffrcststartdt07=frcststartdt07.min
 defadjfrcstmth07 : defadjfrcstmth07= 1( No adjustment) if the implementor
intended for daily batch result to override all forecast adjustment
                    defadjfrcstmth07= 2( Keep Last Change ) if the
implementor intended for previous forecast adjustment to preserve.
                    defadjfrcstmth07= 4( Keep Last Baseline ) if the
implementor intended for previous baseline adjustment to preserve.
                    defadjfrcstmth07= 5( Keep Last PEAK ) if the
implementor intended for previous PEAK adjustment to preserve.
 adjfrcstmthint07 : adjfrcstmthint07 = measureNA(adjfrcstmthint07)
 adjfrcstmthovr07 : adjfrcstmthovr07 = measureNA(adjfrcstmthovr07)
     adjfrcstmthin07 and adjfrcstmthov07 modification is necessary if
implementor want to keep the same adjustment principle for all timeseries.
 runpreprocess : runpreprocess = false
 runnewitembatch: runnewitembatch = false if new item forecast update
during is week is not need.
                      runnewitembatch = true if new item forecast update
during is week is needed.
```

   The previous expressions should be put in a rulegroup that is run after Step 2a in the **hook_pre_preprocess**.

3. Perform the following activities for Step 3.

   When **c_batchDailyB** is true, restore the forecast parameter values after the daily batch is executable

```
deffrcststartdt07 : deffrcststartdt07=c_deffrcststartdt07
 defadjfrcstmth07 : defadjfrcstmth07=c_defadjfrcstmth07
 adjfrcstmthint07 : adjfrcstmthint07=c_adjfrcstmthint07
 adjfrcstmthovr07 : adjfrcstmthovr07=c_adjfrcstmthovr07
 runpreprocess : runpreprocess = c_runpreprocess
 runnewitembatch: runnewitembatch = c_runnewitembatch
```

   The previous expressions should be put in a rulegroup that is run in the **hook_pre_preprocess**.