Oracle® Retail Replenishment Optimization

Implementation Guide

Release 13.1.1

October 2009

ORACLE®

Oracle Retail Replenishment Optimization Implementation Guide, Release 13.1.1

Primary Author: Melissa Artley

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

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

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

(i) the software component known as **ACUMATE** developed and licensed by Lucent Technologies Inc. of Murray Hill, New Jersey, to Oracle and imbedded in the Oracle Retail Predictive Application Server - Enterprise Engine, Oracle Retail Category Management, Oracle Retail Item Planning, Oracle Retail Merchandise Financial Planning, Oracle Retail Advanced Inventory Planning, Oracle Retail Demand Forecasting, Oracle Retail Regular Price Optimization, Oracle Retail Size Profile Optimization, Oracle Retail Replenishment Optimization applications.

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

(iii) the **SeeBeyond** component developed and licensed by Sun MicroSystems, Inc. (Sun) of Santa Clara, California, to Oracle and imbedded in the Oracle Retail Integration Bus application.

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

(v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by SAP and imbedded in Oracle Retail Store Inventory Management.

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

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

(viii) the software component known as **Style Report™** developed and licensed by InetSoft Technology Corp. of Piscataway, New Jersey, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(ix) the software component known as **DataBeacon™** developed and licensed by Cognos Incorporated of Ottawa, Ontario, Canada, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

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

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

Contents

4 Integration

5 Configuration Considerations

6 Batch Processing

7 Internationalization

List of Tables

List of Figures

Preface

The *Oracle Retail Replenishment Optimization Implementation Guide* describes post-installation tasks that need to be performed in order to bring Replenishment Optimization online and ready for production use.

Audience

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

Related Documents

For more information, see the following documents in the Oracle Retail Replenishment Optimization 13.1.1 documentation set:

- *Oracle Retail Replenishment Optimization Release Notes*
- *Oracle Retail Replenishment Optimization Installation Guide*
- *Oracle Retail Replenishment Optimization User Guide*

Customer Support

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

`https://metalink.oracle.com`

When contacting Customer Support, please provide the following:

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

Review Patch Documentation

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

Oracle Retail Documentation on the Oracle Technology Network

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

`http://www.oracle.com/technology/documentation/oracle_retail.html`

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

Conventions

The following text conventions are used in this document:

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

1
Introduction

The primary goal of Replenishment Optimization (RO) is to harness the replenishment methods available in the client's replenishment system. To make best use of the available replenishment capabilities, RO recommends optimized replenishment parameters at the SKU/location level. The recommendations take into account sales volume, volatility, availability of forecast data, seasonality, client business rules and constraints, and financial objectives to determine the optimized values.

RO automatically monitors item/location demand and supply chain variables to determine optimal inventory for greatest return. It recommends replenishment settings, either automatically approving the changes or raising alerts, for example, alerting higher impact items. The optimal replenishment settings recommended by Oracle Retail Replenishment Optimization may be used to update Oracle Retail Advanced Inventory Planning (AIP) replenishment parameters or the retailer's legacy replenishment system. For a more detailed overview of the functionality within RO, see the *Oracle Retail Replenishment Optimization User Guide*.

Contents of this Guide

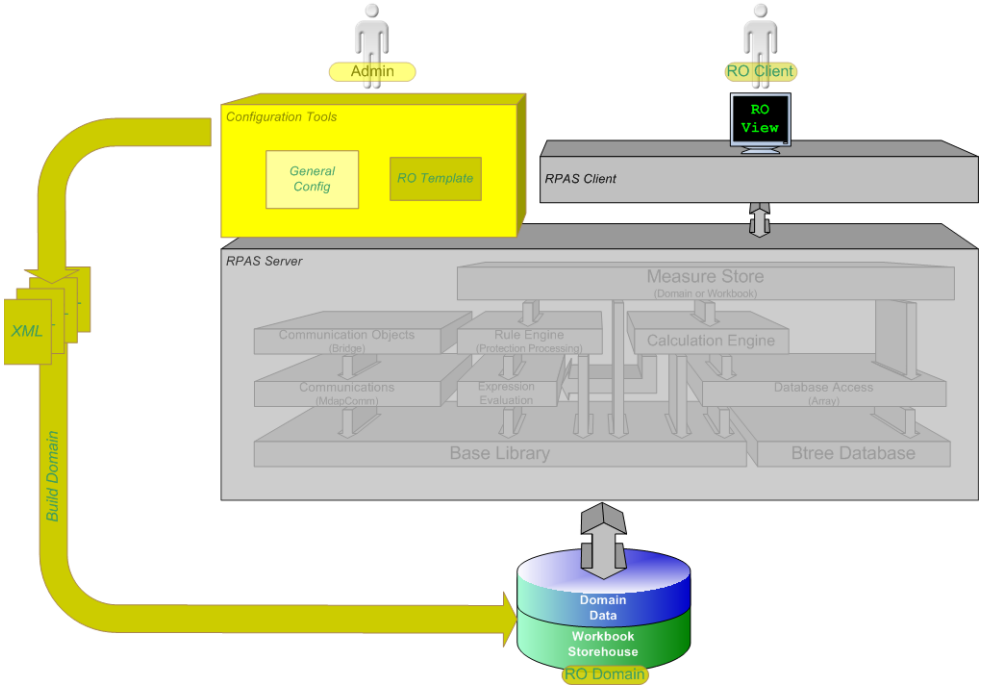
This implementation guide addresses the following topics:

- Chapter 1, "Introduction": Overview of the RO business workflow and skills needed for implementation.
- Chapter 2, "Implementation Considerations": Implementation Considerations. Explanation of the factors to take into consideration before performing the implementation.
- Chapter 3, "Build Scripts": Information on building and patching the RO RPAS domain.
- Chapter 4, "Integration": Overview of the Integrated Inventory Management Suite integration, a detailed review of the Analytic Parameter Calculator for Replenishment Optimization (APC RO) integration, and integration configuration.
- Chapter 5, "Configuration Considerations": Information on the functional changes or enhancements that can be made for RO.
- Chapter 6, "Batch Processing": Explanation of batch scheduling and batch designs.
- Chapter 7, "Internationalization": Translations provided for RO.

RO and the Oracle Retail Enterprise

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

Figure 1–1 RO and the Oracle Retail Enterprise



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

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

- Measures
- Special expressions
- Rules
- Workbook layout

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

Once the domain has been created, the end user can access the domain data through the RPAS client.

Business Process Workflows

Figure 1–2 shows the workflow for the full mode batch of RO. This should be performed when RO is first configured or when a major change to the replenishment plan is necessary.

Figure 1–2 Full Mode Batch Business Process Workflow

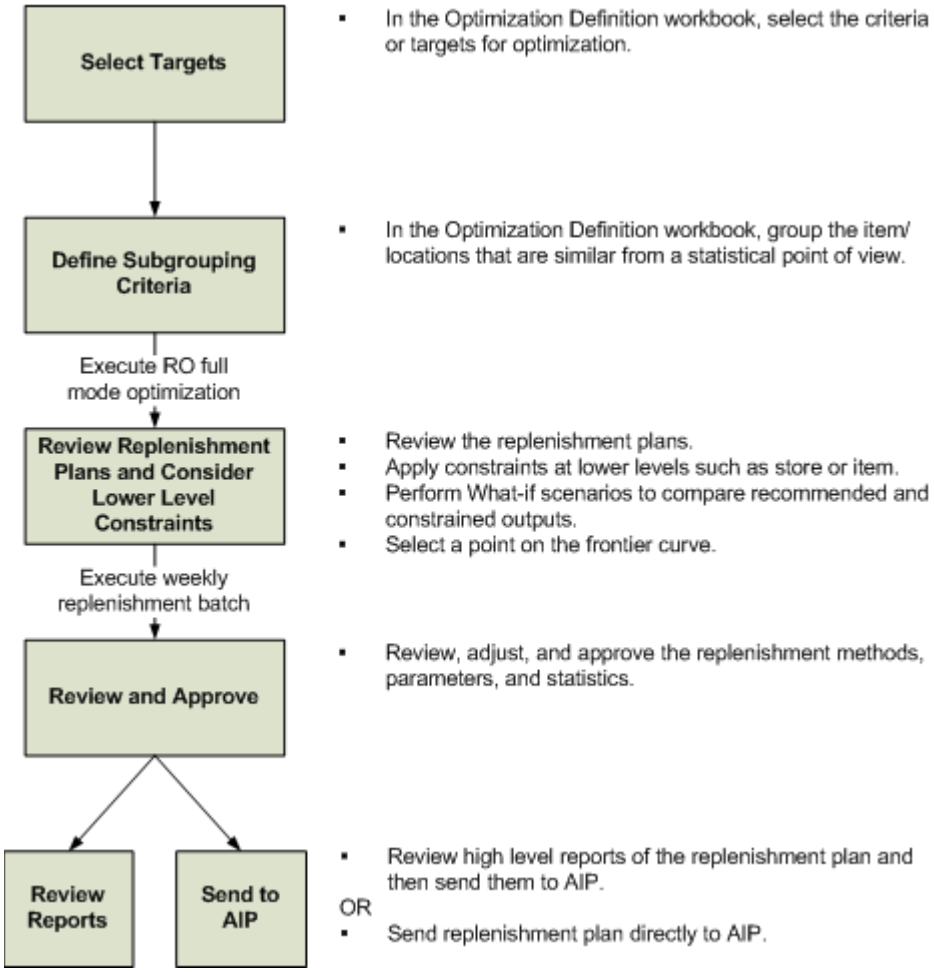
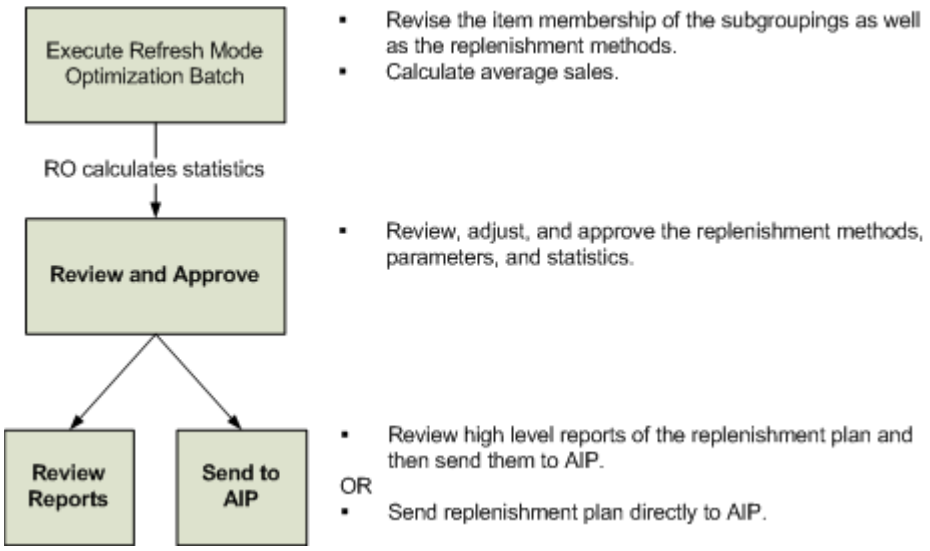


Figure 1–3 shows the workflow for the weekly batch of RO.

Figure 1–3 Weekly Batch Business Process Workflow



Skills Needed for Implementation

The implementer needs an understanding of the following applications and technical concepts.

Applications

The implementer should understand the interface requirements of the integrated applications and data sources for the master data, demand, and inventory history. For the Integrated Inventory Management Suite, the implementer needs this knowledge for the following applications:

- Oracle Retail Analytic Parameter Calculator Replenishment Optimization
- Oracle Retail Advanced Inventory Planning
- Oracle Retail Demand Forecasting

Technical Concepts

The implementer should understand the following technical concepts:

- UNIX system administration, shell scripts, and job scheduling
- Performance constraints based on the retailer's infrastructure
- Technical architecture for RO
- Retailer's hierarchical (SKU/store/day) data
- RO batch processes
- How to set up an RPAS domain
- A basic understanding of RPAS configuration and how to use the RPAS Configuration Tools
- Understanding of how RPAS rule language works

- Understanding of measures and dimension constructs
- Basic merchandising
- Basic forecasting

2

Implementation Considerations

The following information needs to be considered before implementing RO:

- Historical Data
- Domain Partitioning
- Formatting
- Plug-ins
- Patch Considerations
- Batch Scheduling
- Security
- Internationalization

Historical Data

It is recommended that you have at least two years of historical sales and inventory data for creating replenishment optimization plans. Less data can be used, but the more data that is available, the more statistical significance can be given to replenishment optimization.

Domain Partitioning

Partitioning is done to avoid contention for resources. Building a workbook and committing data are two processes that can cause contention.

How data is partitioned has an impact on the business process. The RO domain is defined as a global domain. For performance reasons, a simple domain is not recommended. There should be an even distribution of users across a set of local domains. For example, men's merchandise could be in a domain, women's merchandise in a domain, and children's merchandise in a domain. When a user is committing data in the men's merchandise domain, this will not affect the users in the women's or children's domains because of the use of partitioning.

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

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

Domain partitioning is supported on any dimension of the Product (PROD) hierarchy and any dimension of the Location (LOC) hierarchy if the optimization level is below partition. These hierarchies are standard RPAS hierarchies.

Note: The partitioning level in the RO configuration is PGRP, which is labeled Group. It is recommended that this not be changed.

In the GA configuration, Group is a dimension label. The group dimension is a regular dimension in the product hierarchy, which the customer can rename or delete. One of the major purposes of partitioning in RO is to facilitate the parallelization of the batch process.

Formatting

Formatting can be done in the configuration or the workbook after the domain is built.

- An implementer can create generic styles for the measures and assign them to measure components or realized measures. For each measure, these styles can be overridden on each workbook template. Formatting can only be changed by using the RPAS Configuration Tools. For more information, see the *Oracle Retail Predictive Application Server Configuration Tools User Guide*.
- Once the domain is built, the implementer can set up worksheet sizes and placements, exception value formatting, gridlines, and other formatting. The implementer instantiates a workbook of the template to set up specific formatting by using the Format menu. The updated format is then saved to the template so that it is available to all users for any newly created workbooks. For information on how to use the Format menu, see the *Oracle Retail Predictive Application Server User Guide*.

Plug-ins

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

Plug-ins are application-specific Java code modules that run inside and automate the RPAS Configuration Tools to assist the implementer with specific application configuration. There are rules that an implementer must follow when configuring an application. A plug-in makes such adherence easier by automating parts of the configuration process and validity-checking entries that are made.

Patch Considerations

With a new release, there are two types of patches that can affect the RO RPAS domain:

- Changes to the code in the RPAS libraries

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

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

Batch Scheduling

Batch scripts are lists of commands or jobs executed without human intervention. A batch window is the time frame in which the batch process must run. It is the upper limit on how long the batch can take. Batch scripts are used for importing and exporting data and for generating size profiles. The retailer needs to decide the best time for running batch scripts within the available batch window.

How often to upload updated sales and inventory data needs to be determined.

Security

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

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

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

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

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

Internationalization

For more information on translation for RO, see Chapter 7, "Internationalization".

3

Build Scripts

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

Installation Dependencies

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

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

Environment Setup

Before downloading the installation package to the UNIX server, a central directory structure to support the environment needs to be created. This central directory is referred to in this document as `<ro_directory>`. The UNIX user performing the installation needs to set up an environment variable called `RO_HOME` in the user's .profile:

export RO_HOME=<*full path name to RO home*>

RPAS Installation

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

The RPAS installer performs the following functions:

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

RO Installation

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

The RO installer performs the following functions:

- Downloads the configuration and batch scripts into the `<ro_directory>/Config` and `<ro_directory>/bin` directories
- Downloads a set of sample hierarchy and data files into the `<ro_directory>/TestInput` directory
- Build a sample domain at `<ro_directory>/domain/RO`

Custom Domain Build

To do a custom build of a domain:

1. Update the globaldomainconfig.xml file with the correct domain paths.
2. If needed, update the default environment variables in ro_environment.sh.
3. Execute the buildRO.sh script:

   ./buildRO.sh

After the first time buildRO.sh is executed, an error may occur when it tries to remove the old log file because a log file does not yet exist.

Environment Variables

In addition to the regular RPAS environment variables, including RPAS_HOME, you need to set up the following environment variables and export them:

- **RPAS_JAVA_CLASSPATH**

  This is the path to the RO JAVA libraries. It should appear as follows: "$RPAS_HOME/applib/aaiReplenOpt.jar;$RPAS_HOME/applib/aaijni.jar"

  Note: For non-Windows platforms, use a colon instead of a semicolon in RPAS_JAVA_CLASSPATH.

- **RIDE_HOME**
  This is the full directory path to where ConfigTools was installed.
- **RO_HOME**
  This is the central directory structure to support the environment.
- **RO_DOMAIN**
  This is the full directory path of your domain.
- **PATH**
  Include $RPAS_HOME/bin, $RO_HOME/bin, and $RIDE_HOME/bin in the path.

Files Needed to Build the RO - RPAS Domain

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

- Standard RPAS hierarchy files
- RO-specific hierarchy files
- Data files

Standard RPAS Hierarchy Files

The following hierarchy files are needed:

- Calendar (CLND) Hierarchy File
- Product (PROD) Hierarchy File
- Location (LOC) Hierarchy File
- Warehouse (WHS) Hierarchy File

> **Note:** As with all standard RPAS hierarchies, these hierarchies are configurable. For information about configuring these hierarchies, see Chapter 5, "Configuration Considerations".

Calendar (CLND) Hierarchy File

File name: clnd.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–1 Calendar Hierarchy Fields

Field	Description
DAY	Day ID
DAY label	Day label
WEEK	Week ID
WEEK label	Week label
MNTH	Fiscal Month ID
MNTH label	Fiscal Month label
QRTR	Fiscal Quarter ID
QRTR label	Fiscal Quarter label
SSN	Fiscal half ID
SSN label	Fiscal half label
YEAR	Fiscal Year ID
YEAR label	Fiscal Year label
DOW	Day of the Week ID
DOW label	Day of the Week label
DOS	Day of the Season ID
DOS label	Day of the Season label

Table 3–1 (Cont.) Calendar Hierarchy Fields

Field	Description
WOY	Week of the Year ID
WOY label	Week of the Year label
WOS	Week of the Season ID
WOS label	Week of the Season label

Example:

```
20091230,12/30/2009,W53_2009,1/1/2010,DEC_2009,December, FY
2009,Q4_2009,Quarter 4, FY 2009,S4_2009,Season 4, FY
2009,A2009,FY2009,WED,Wednesday,DOS96,DOS 96,WY53,Week
53,WS14,WOS 14
```

Product (PROD) Hierarchy File

File name: prod.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–2 Product Hierarchy Fields

Field	Description
ITEM	SKU ID
ITEM label	SKU label
ITPT	Item Parent ID
ITPT label	Item Parent label
ITGR	Item Group ID
ITGR label	Item Group label
SCLS	Subclass ID
SCLS label	Subclass label
CLSS	Class ID
CLSS label	Class label
DEPT	Department ID
DEPT label	Department label
PGRP	Group ID
PGRP label	Group label
DVSN	Division ID
DVSN label	Division label
SPLR	Supplier ID
SPLR label	Supplier label
PTD1	Parent Diff1 ID
PTD1 label	Parent Diff1 label
GPD1	Grand Parent Diff1 ID

Table 3–2 (Cont.) Product Hierarchy Fields

Field	Description
GPD1 label	Grand Parent Diff1 label
SCD1	Subclass Diff1 ID
SCD1 label	Subclass Diff1 label
CLD1	Class Diff1 ID
CLD1 label	Class Diff1 label
DPD1	Department Diff1 ID
DPD1 label	Department Diff1 label
DIF1	Diff1 ID
DIF1 label	Diff1 label

Example:

```
10772144,LIP COLOR- 31RED:31RED:NONE,10772143,LIP COLOR-
31RED,10182143,LIP COLOR-
31RED,310,STODA,310,STODA,310,STODA,4500,Group 5,1,All
Product,543213759,NORDELL,10182143,LIP COLOR- 31RED,10182143,LIP
COLOR- 31RED,310,STODA,310,STODA,310,STODA,_sml,Small
```

```
10184464,LIP ROUGE- 01PUCKER:01PUCKER:NONE,10184463,LIP ROUGE-
01PUCKER,10182163,LIP ROUGE-
01PUCKER,310,STODA,310,STODA,310,STODA,4500,Group 5,1,All
Product,553213760,NORDELL,10182163,LIP ROUGE-
01PUCKER,10182163,LIP ROUGE-
01PUCKER,310,STODA,310,STODA,310,STODA,_sml,Small
```

Location (LOC) Hierarchy File

File name: loc.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–3 Location Hierarchy Fields

Field	Description
STR	Store ID
STR label	Store label
DSTR	District ID
DSTR label	District label
REGN	Region ID
REGN label	Region label
AREA	Area ID
AREA label	Area label
CHN	Chain ID
CHN label	Chain label
CMPN	Company ID
CMPN label	Company label
SFMT	Store format ID
SFMT label	Store Format label
STCL	Store class ID
STCL label	Store class label

Example:

```
769,store number 769,769,store number 769,3,region number
3,3,region number 3,3,region number 3,200,Company 1,0,0,A,A

771,store number 771,771,store number 771,3,region number
3,3,region number 3,3,region number 3,200,Company 1,0,0,A,A
```

Warehouse (WHS) Hierarchy File

File name: whs.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–4 Warehouse Hierarchy Fields

Field	Description
WRHS	Warehouse ID
WRHS label	Warehouse label

Example:

```
east, east warehouse
west, west warehouse
```

RO Hierarchy Files

The following are required hierarchy files needed for RO:

- Subgroup Hierarchy File
- APC Group Hierarchy File
- Scenario Hierarchy File
- Frontier Data Point Hierarchy File
- Break Point Hierarchy File
- KEK Hierarchy File
- PI Hierarchy File

Subgroup Hierarchy File

The subgroup hierarchy represents the subgroups generated in the statistical/breakpoint grouping. The number of subgroups that can be generated must be within 1 and the subgroup dimension length. If you want to increase the number of subgroups beyond that limit, you must prepare a subgroup hierarchy file and load the hierarchy with that new file. This hierarchy contains one dimension: subgroup.

File name: subg.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–5 Subgroup Hierarchy Fields

Field	Description
SGRP	Subgroup ID
SGRP label	Subgroup label

Example:

```
48, subgroup 48
49, subgroup 49
```

APC Group Hierarchy File

The APC group hierarchy represents the statistical groups generated by APC RO using a statistical grouping. The APC group dimension length should be decided by the APC RO calculation. This hierarchy contains one dimension: APC group.

File name: apcg.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–6 APC Group Hierarchy Fields

Field	Description
AGRP	APC Group ID
AGRP label	APC Group label

Example:

```
48, APC group 48
49, APC group 49
```

Scenario Hierarchy File

The scenario hierarchy represents the different replenishment scenarios used in APC RO. The scenario dimension length should be decided in the APC RO setup. This hierarchy contains one dimension: scenario.

File name: scn.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–7 Scenario Hierarchy Fields

Field	Description
SCNO	Scenario ID
SCNO label	Scenario label

Example:

```
48, scenario 48
49, scenario 49
```

Frontier Data Point Hierarchy File

The frontier data point hierarchy represents the frontier data point on the Inventory/Service Level Trade-off curve that is generated by optimization. The number of frontier data points in a calculation = (scenario -1) * subgroup number. This hierarchy contains one dimension: frontier data points.

File name: fcdp.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–8 Frontier Data Point Hierarchy Fields

Field	Description
FCPT	Frontier data point ID
FCPT label	Frontier data point label

Example:

```
00198,points 00198
00199,points 00199
```

Break Point Hierarchy File

The break point hierarchy represents the break points used in the user-defined break point grouping. The number of position names in the break point dimension should be larger than the number of groups per grouping factor. This hierarchy contains one dimension: break points.

File name: brkp.csv.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–9 Break Point Hierarchy Fields

Field	Description
BKPT	Break point ID
BKPT label	Break point label

Example:

```
08, break point 8
09, break point 9
```

KEK Hierarchy File

The KEK hierarchy represents the data points in the w-to-z lookup table used in the dynamic replenishment calculation. The KEK dimension length is decided by the number of entries in the w-to-z lookup table. This hierarchy contains one dimension: KEK.

File name: kek.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–10 KEK Hierarchy Fields

Field	Description
EW	KEK ID
EW label	KEK label

Example:

```
204,e_204
```

PI Hierarchy File

The PI hierarchy represents the data points in the poisson safety stock lookup table used in poisson replenishment calculation. The PI dimension is decided by the number of entries in the lookup table. This hierarchy contains one dimension: PI.

File name: pi.dat
File format: comma-separated values file

The following table describes the fields in the file:

Table 3–11 Demand Group Hierarchy Fields

Field	Description
PIK	PI ID
PIK label	PI label

Example:

```
pi098,pi_98
pi099,pi_99
```

Required Data Files

The following data files are required:

Measures from a Merchandising or Replenishment System

- **Item/store/week**
  - **Weekly sales**: wklydemandu.csv.ovr
  - **Weekly on hand inventory**: wklyinv_ohu.csv.ovr
  - **Weekly on order inventory**: wklyinv_oou.csv.ovr
  - **Weekly order unit**: wklyorderu.csv.ovr
  - **Weekly lost sales units**: wklylostslsu.csv.ovr
  - **Weekly out of stock units**: wklyoostku.csv.ovr
- **Item/store**
  - **Price**: gprice_.csv.ovr
  - **Cost**: gcost_.csv.ovr
  - **Packsize**: gpacksize_.csv.ovr
  - **PresentationStock**: gpres_stock.csv.ovr
  - **Leadtime**: gleadtime_.csv.ovr
  - **Reviewtime**: greview_time_.csv.ovr
  - **Rounding threshold**: ground_thrsh_.csv.ovr
  - **Space per unit**: spaceis.csv.ovr

Measures from a Demand Forecasting Application

- **Item/store/week**
  - **Weekly forecast**: wklyfcstu.csv.ovr
  - **Weekly cumint**: wklycumint_.csv.ovr

Measures from APC RO

- **Scenario**
  - **Replenishment method**: rplmthscn.csv.ovr, string type
  - **Replenishment parameter 1**: rplparm1scn.csv.ovr
  - **Replenishment parameter 2**: rplparm2scn.csv.ovr
  - **Replenishment auxiliary parameter 1**: auxparm1scn.csv.ovr
  - **Replenishment auxiliary parameter 2**: auxparm2scn.csv.ovr
  - **Scenario priority**: priorityscn.csv.ovr
- **APC group**
  - **APC group leadtime lower bound**: apcltlbag.csv.ovr
  - **APC group leadtime upper bound**: apcltubag.csv.ovr
  - **APC group mean sales lower bound**: apcmeanlbag.csv.ovr
  - **APC group mean sales upper bound**: apcmeanubag.csv.ovr

  - **APC group sales variability lower bound**: apcvarlbag.csv.ovr
  - **APC group sales variability upper bound**: apcvarubag.csv.ovr
- **APC group/scenario**
  - **Service level**: servlevelgs.csv.ovr
  - **Simulated average demand**: simavgdmndgs.csv.ovr
  - **Week of supply**: wosgs.csv.ovr
  - **Average order frequency**: avgordfrqgs.csv.ovr
  - **Average order size**: avgordsizgs.csv.ovr
  - **Wastage**: simwastegs.csv.ovr
  - **Stockouts**: stkoutnmgs.csv.ovr
- **Item/store/scenario**
  - **Service level**: servleveliss.csv.ovr
  - **Simulated average demand**: simavgdmndiss.csv.ovr
  - **Weekly of supply**: wosiss.csv.ovr
  - **Average order frequency**: avgordfrqiss.csv.ovr
  - **Average order size**: avgordsiziss.csv.ovr
  - **Wastage**: simwasteiss.csv.ovr
  - **Stockouts**: stkoutnmiss.csv.ovr

Internal Data

- **KEK**
  - **W value**: gw_val.ovr
  - **Z value**: gz_val_.ovr
- **PI**
  - **Average sales index**: gmean_sls_ind.ovr
  - **Safety stocks**: gpi_val.ovr

Building the RO - RPAS Domain

The script use to build or patch the RO - RPAS domain is described in this section. The script is located in the `<ro_directory>/bin` directory.

Note: To patch a domain, use buildRO.sh with the -p flag as described in the table below.

Batch Designs

This section contains detailed information on the following build scripts:

- Building a Domain

Building a Domain

Script
buildRO.sh

Usage
buildRO.sh <options -cdil> <flags -gpt>

Argument	Allowed Values	Description
options	c	Configuration directory Default is `<ro_directory>/config`
	d	Domain path Default is `<ro_directory>/domain`
	i	Input directory Default is `<ro_directory>/input`
	l	Log directory Default is `<ro_directory>/logs`
flags	g	Set this flag to use debug function libraries
	p	Set this flag to make a patch build
	t	Set this flag to make a test build

Notes

- The script uses the Configuration Tools rpasInstall utility to build a domain. See the *Oracle Retail Predictive Application Server Administration Guide* for details on this utility.
- The script also uses the mace and loadmeasure RPAS utilities. See the *Oracle Retail Predictive Application Server Administration Guide* for details on these utilities.
- All hierarchy and measure files are placed in $RO_HOME/input.
- The script creates two initial users: adm and usr. The passwords are initially set to be the same as the user ID.

Loading and Extracting Data

Data is loaded into RO using the standard RPAS approach. See the *Oracle Retail Predictive Application Server Administration Guide* for details on formatting the load data files and on the utilities that enable administrators to load data into RPAS. For information on integration scripts, see Chapter 4, "Integration". For information on any other batch scripts, see Chapter 6, "Batch Processing".

4

Integration

This chapter describes the following information:

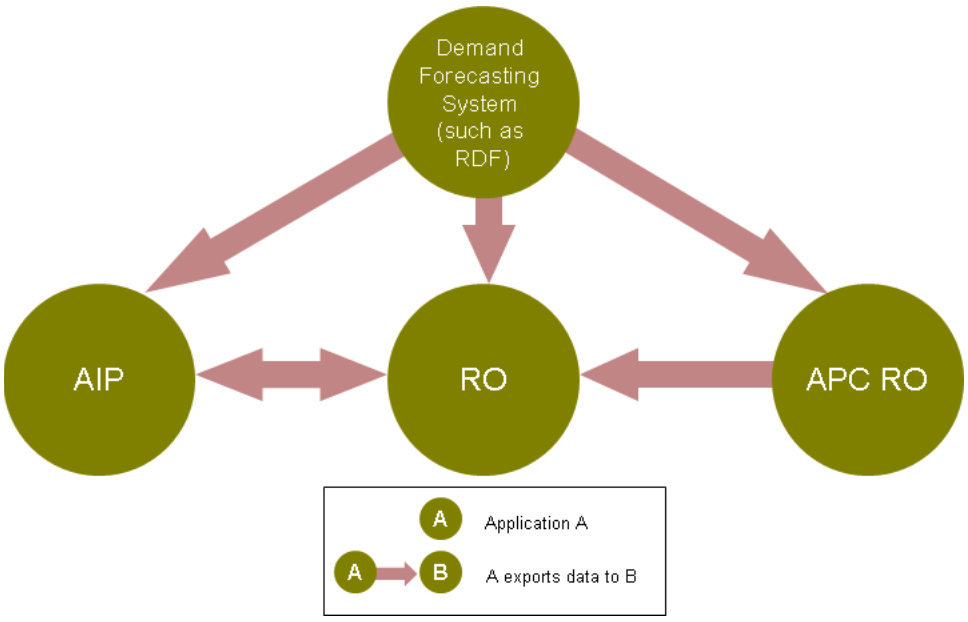
- Overview of the Integrated Inventory Management Suite
- Overview of the APC RO and RO Integration
- Integration Configuration

Overview of the Integrated Inventory Management Suite

The Integrated Inventory Management Suite is the integration of Demand Forecasting (RDF), Advanced Inventory Planning (AIP), Replenishment Optimization (RO), and Analytic Parameter Calculator Replenishment Optimization (APC RO) as a full-suite inventory management solution for retailers.

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

Figure 4–1 Conceptual Overview



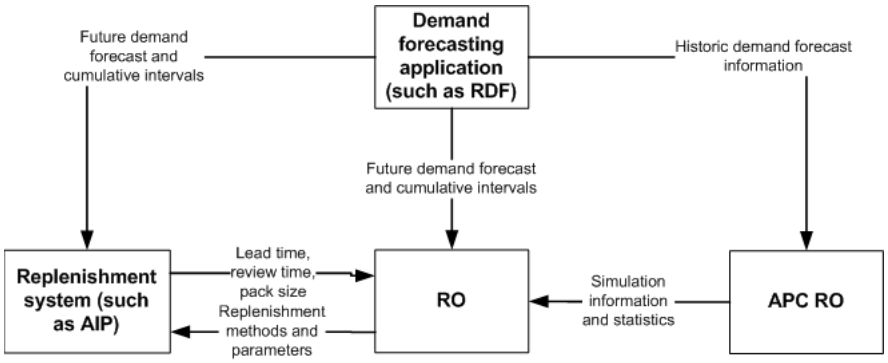
Note: The integration interface between RDF and AIP is not fully certified.

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

Integrated Inventory Management Suite Data Flow

Figure 4–2 shows the integration of the Integrated Inventory Management Suite applications and the flow of data among those applications. Note that the figure below shows a replenishment system. This can be AIP or any other replenishment system. The demand forecasting application can be RDF or any other forecasting system.

Figure 4–2 Integrated Inventory Management Suite Data Flow



Note: The integration interface between RDF and AIP is not fully certified.

Data Flow Description

These descriptions explain each of the data flows in Figure 4–2.

From a Demand Forecasting Application (such as RDF) to APC RO

- Sends historic demand forecasts for a forecast horizon for a series of forecast start dates. It sends a separate forecast file for each forecast start date.

From a Demand Forecasting Application (such as RDF) to a Replenishment System (such as AIP)

- Sends time-phased demand forecasts (starting today and looking forward) at the item/store level.
- Sends the cumulative standard deviation of the forecast. This is needed for the calculation of safety stock.

From APC RO to RO

- Sends simulation information and statistics:
  - Item/location/scenario level information
  - Mean/variability/lead time grouping level information
  - Mean/variabilitygrouping/lead time/scenario level information
  - Scenario level information

From a Replenishment System (such as AIP) to RO

- Sends the lead time in flat files. The lead time (or order cycle) pattern generally contains the same lead time on all days that have a lead time; however, the lead time may increase for the weekend. Therefore the most common lead time is found during the business week.
- Sends the review time in flat files. Review time is the number of days until the next possible receipt. It is a key factor in determining the minimum amount of projected stock that should be available until the next receipt. Since review time can change daily, the minimum available inventory must cover the longest review time in order to avoid stock outs.
- Sends the ordering pack size in flat files. The ordering pack size is the preferred pack size of an item that should be ordered from a source to the destination.

From a Demand Forecasting Application (such as RDF) to RO

- Sends time-phased demand forecasts (starting at the current day and looking forward) at the item/store level. This allows the user to understand how the replenishment settings would perform based on that demand.
- Sends the cumulative standard deviation of the forecast. This is needed for the calculation of safety stock.

From RO to a Replenishment System (such as AIP)

- Sends the recommended replenishment methods and parameters in flat files based on the schedule that the user sets.
- For AIP specifically, RO performs the necessary transformations needed to convert order-based replenishment parameters to a relevant form before sending it to AIP since AIP is a receipts-based system.

Overview of the APC RO and RO Integration

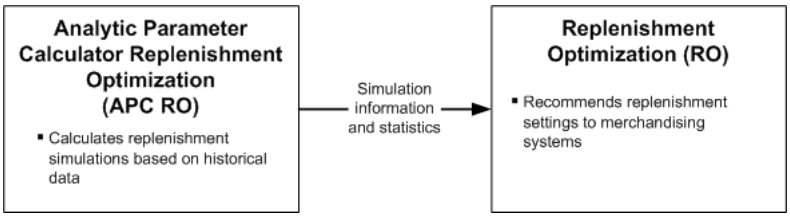
APC RO is an analytical, fact-based application that uses a client's historical sales patterns to perform replenishment simulations and calculate statistics. RO uses the APC RO results to make optimal replenishment recommendations based on specific business goals and retail constraints.

See the APC RO documentation for more information.

APC RO and RO Data Flow

Figure 4–3 shows the integration of APC RO and RO and the flow of data between them.

Figure 4–3 APC RO and RO Integration



Data Flow Description

These descriptions explain the data flow in Figure 4–3.

From APC RO to RO

The following data is imported into RO from APC RO:

- Item/location/scenario level information

Table 4–1 Item/Location/Scenario Level Information

Name	ETL Name
Store Weeks of Supply Data	wosiss.csv.ovr
Store Service Level Data	servleveliss.csv.ovr
Store Lead Time Data	gleadtime_.csv.ovr
Store Simulated Average Demand Data	simavgdmndiss.csv.ovr
Store Average Order Frequency Data	avgordfrqiss.csv.ovr
Store Average Order Size Data	avgordsiziss.csv.ovr
Store Number of Stock-outs Data	stkoutnmiss.csv.ovr
Store Average Simulated OUTL Data	simavgoutliss.csv.ovr
Store Average Simulated OP Data	simavgopiss.csv.ovr
Store Average Wastage Data	simwasteiss.csv.ovr

- Mean/variability/lead time grouping level information

Table 4–2 Mean/Variability/Lead Time Grouping Level Information

Name	ETL Name
Mean Lower Bound Data	apcmeanlbag.csv.ovr
Mean Upper Bound Data	apcmeanubag.csv.ovr
Variability Lower Bound Data	apcvarlbag.csv.ovr
Variability Upper Bound Data	apcvarubag.csv.ovr
Lead Time Lower Bound Data	apcltlbag.csv.ovr
Lead Time Upper Bound Data	apcltubag.csv.ovr

- Mean/variability grouping/lead time/scenario level information

Table 4–3 Mean/Variability Grouping/Scenario Level Information

Name	ETL Name
Store G/S Weeks of Supply Data	wosgs.csv.ovr
Store G/S Service Level Data	servlevelgs.csv.ovr
Store G/S Simulated Average Demand Data	simavgdmndgs.csv.ovr
Store G/S Average Order Frequency Data	avgordfrqgs.csv.ovr
Store G/S Average Order Size Data	avgordsizgs.csv.ovr
Store G/S Number of Stock-outs Data	stkoutnmgs.csv.ovr
Warehouse G/S WOS Data	whswosgw.csv.ovr
Warehouse G/S Simulated Average Issues Data	whssimdmdgw.csv.ovr
Store G/S Average Simulated OUTL Data	simavgoutlgs.csv.ovr
Store G.S Average Simulated OP Data	simavgopgs.csv.ovr
Store G/S Average Wastage Data	simwastegs.csv.ovr

Scenario level information

Table 4–4 Scenario Level Information

Name	ETL Name
Store Replenishment Method Data	rplmthscn.csv.ovr
Store Parameter 1 Data	rplparm1scn.csv.ovr
Store Parameter 2 Data	rplparm2scn.csv.ovr
Store Auxiliary Parameter 1 Data	auxparm1scn.csv.ovr
Store Auxiliary Parameter 2 Data	auxparm2scn.csv.ovr

Integration Configuration

The following sections describe the configuration needed for RO:

- Setting Environment Variables
- Setting Default Measure Values
- Integration Scripts

- Batch Environment Scripts
- Using runBatch
- Batch Designs
- RO Stores Data Mapping

Setting Environment Variables

After RO is installed, define the environment variables that RO needs to execute.

ro_environment.sh

The variables displayed in Table 4–5 need to be defined properly.

Table 4–5 Environment Variables

Environment Variable	Description
RO_DOMAIN	The location of the RO domain. The default is set to the value set in TEST_RO_DOMAIN.
RO_CONTROL_FILES	The location of control files for this interface. The default is set to ${RO_DOMAIN}/control_files.
RO_TEMP_DIR	This should be a writable path. Use of /tmp is discouraged due to the typical small size of this partition on UNIX machines.
RO_LOG_HOME	The location of log files. The default is set to ${RO_DOMAIN}/logs.
RO_LOG_LEVEL	The default is set to ERROR if $RPAS_LOG_LEVEL is not specified.
RO_FILE_TYPE	Format of files can be either CSV or TXT. The default is set to CSV.
AIP_INPUT	The location of input files coming from AIP. The default is ${RO_DOMAIN}/interface/aip/input.
AIP_OUTPUT	The location of output files to be sent to AIP. Default is ${RO_DOMAIN}/interface/aip/output.
CLASSPATH	Include the following in CLASSPATH: "$RPAS_HOME/applib/RO.jar" "$RPAS_HOME/applib/groovy-all.jar" "$RPAS_HOME/applib/ant.jar" "$RPAS_HOME/applib/ant-launcher.jar" "$RPAS_HOME/applib/aaiReplenOpt.jar" "$RPAS_HOME/applib/aaijni.jar"

Setting Default Measure Values

The following measures need to be adjusted to have the correct values based on implementation.

Export to Replenishment System

This measure specifies which product/location should be considered when extracting data for AIP. By default all product/location data will be exported if the measure is not populated.

Integration Scripts

Integration scripts are used for moving data between applications. Table 4–6 lists the integration scripts for RO.

Table 4–6 RO Integration Scripts

Application	Script	Arguments	Description
RO Stores Implementation	ro_export_to_aip.sh	None	This is the main script that invokes the export process and creates data files to be sent to AIP.
RO Stores Implementation	ro_import_from_aip.sh	None	This is the main script that invokes the load process and loads AIP data into RO.

Batch Environment Scripts

ro_environment.sh sets the necessary environment variables for execution of RO integration scripts. It is invoked within the integration scripts listed in the table above. The details of environment variables are listed in the Setting Environment Variables section.

Using runBatch

To execute a batch, the integration scripts in RO make a call to the Groovy class runBatch which parses the xml file <batchStep>.xml.

The following describes the Groovy runBatch.class:

- runBatch is invoked like any other Java class provided that the Groovy jar files have been properly set in the class path.
- runBatch requires two arguments that are paths to a systemVariable.xml which is located in the RPAS_HOME. The other XML file contains details of the batch to run. For instance, for RO to export to AIP, use the xml file called exportSteps.xml. For RO import from AIP, use a similar xml file called importSteps.xml.

  ```
  java runBatch $RPAS_HOME/applib/resources/systemVariables.xml $RO_
  DOMAIN/interface/aip/<   >.xml
  ```

- <batchStep>.xml should be located at $RO_DOMAIN/interface/aip
- <batchStep>.xml follows the format below. Within the batchSteps tags there are repeated batchStep tags which represent the different steps in the batch. Each batchStep has two attributes: name and run. To turn off a specific batchStep, change the run attribute from **true** to **false**.

```
<batchsteps>
  <batchstep>
```

```
    <name>Step1</name>
    <run>true</run>
  </batchstep>
  <batchstep>
    <name>Step2</name>
    <run>true</run>
  </batchstep>
</batchsteps>
```

Batch Designs

This section contains detailed information on the following integration scripts:

- ro_export_to_aip.sh
- ro_import_from_aip.sh

ro_export_to_aip.sh

For each local domain associated with RO_DOMAIN, ro_export_to_aip.sh calls runBatch with the following arguments:

```
java runBatch "$RPAS_HOME/applib/resources/systemVariables.xml"
"{local-domain}/interface/aip/mainCalcSteps.xml"
```

It then calls runBatch with the following arguments:

```
java runBatch "$RPAS_HOME/applib/resources/systemVariables.xml"
"$RO_DOMAIN/interface/aip/mainExportSteps.xml"
```

systemVariables.xml

Path: $RPAS_HOME/applib/resources

For systemVariables.xml, values within the tags contain environment variables in which the attribute mentioned is stored.

```
<variables>
  <global-domain>RO_DOMAIN</domain-path>
  <log-level>RO_LOG_LEVEL</log-level>
  <log-home>RO_LOG_HOME</log-home>
</variables>
```

mainCalcSteps.xml

Path: $RO_DOMAIN/interface/aip

Batchsteps:

- calcMinMax
- calcTimeSupply
- calcDynamic
- calcHybrid
- calcPoisson
- calcMinSafetyStock
- calcRoUpdate

Table 4–7 Batch Steps for exportSteps.xml

Batch Step	Description
calcMinMax	Calculates all measures required for STR_MINMAX export.
calcTimeSupply	Calculates all measures required for STR_TIMESUPPLY export.
calcDynamic	Calculates all measures required for STR_DYNAMIC export.
calcHybrid	Calculates all measures required for STR_HYBRID export.
calcPoisson	Calculates all measures required for STR_POISSON export.
calcMinSafetyStock	Calculates all measures required for STR_MINSS export.

mainExportSteps.xml

Path: $RO_DOMAIN/interface/aip

ro_import_from_aip.sh

ro_import_from_aip.sh calls runBatch with the following arguments:

```
java runBatch $RPAS_HOME/applib/resources/systemVariables.xml $RO_
DOMAIN/interface/aip/exportSteps.xml
```

systemVariables.xml

Path: $RPAS_HOME/applib/resources

For systemVariables.xml, values within the tags contain environment variables in which the attribute mentioned is stored.

```
<variables>
  <global-domain>RO_DOMAIN</global-domain>
  <log-level>RO_LOG_LEVEL</log-level>
  <log-home>RO_LOG_HOME</log-home>
</variables>
```

importSteps.xml

Path: $RO_DOMAIN/interface/aip

The import batch has a single step.

Batchstep:

- loadAipData

Table 4–8 Batch Step for importSteps.xml

Batch Step	Description
loadAipData	Loads file STR_AIP_DATA which is generated from AIP into RO measures. If RO_FILE_TYPE is txt, a fixed width file named STR_AIP_DATA.txt is required for this step. However, if the setting for RO_FILE_TYPE is CSV, file STR_AIP_DATA.csv will be required.

RO Stores Data Mapping

RO currently only supports stores. The mapping data below is for stores only.

Input Measure Data

The table below displays the measure files that RO sends to AIP.

Table 4–9 Input Measure Data

File Name	Measures Loaded
STR_MINMAX	sr0_minstk_i, sr0_maxstk_i and sr0_rplmtd_i
STR_DYNAMIC	sr0_svclvl_i, sr0_isd_i and sr0_rplmtd_i
STR_TIMESUPPLY	sr0_mints_day_i, sr0_maxts_day_i, sr0_ts_hzn_i and sr0_ rplmtd_i
STR_HYBRID	sr0_mints_day_i, sr0_isd_i and sr0_rplmtd_i
STR_POISSON	sr0_svclvl_i, sr0_isd_i and sr0_rplmtd_i
STR_MINSS	sr0_minss_unt_i
STR_ROUpdate	iproupdtstri

STR_MINMAX

Table 4–10 STR_MINMAX Data Element Details

Data Type	Data Element Name	Data Description
Measure	Store Min Stock	Measure at the SKU/Store/Day level. Used in MIN/MAX Replenishment Method. Min Stock is used in determining the inventory level at which replenishment is triggered.
Measure	Store Max Stock	Measure at the SKU/Store/Day level. Set in MIN/MAX Replenishment Method. The receipt plan quantity is calculated to bring Net Inventory up to the max.
Measure	Store Repl Method	Measure at the SKU/Store/Day level. The Replenishment Method determines the type of calculations that will be used in generating a store replenishment plan. The default is No Replenishment.

Table 4–11 STR_MINMAX Extracting Program Details

Program Language	Korn Shell Script
Program Name	ro_export_to_aip.sh
Files Used	exportMinMax.xml, createFile.xml and transferFilePositions.xml
Program Type	On request

Table 4–12 STR_MINMAX Data Source and Target Details

Data Source Details			Target Data Details	
Data Origin System	RO		Target Object Type	RPAS Measures
Source Object Type	Fixed Length Text File or Comma Separated Values File		Target Object Names	sr0_minstk_i, sr0_maxstk_i and sr0_rplmtd_i
Source Object Name	STR_MINMAX		Target Object Databases	data/sr0_minstk_i, data/sr0_maxstk_i and data/sr0_ rplmtd_i
Required/Optional	Optional		Target Object Load Intersection	day_str_sku_

Table 4–13 STR_MINMAX Field Level Mapping - Source

Source Fields	Source Field Description	Field Start Position	Field Width
aipefctvdate_	Effective date	1	9
STR_	STR	10	20
ITEM	SKU	30	20
aipreplpval1_	Replenishment Parameter 1	50	10
aipreplpval2_	Replenishment Parameter 2	60	10
aiprplmtd_	Replenishment Method	80	10

Table 4–14 STR_MINMAX Field Level Mapping - Target

Target Data Field Name	Target Field Description	Target Field Data Type	Condition/Format
DAY_	DAY	String	"D20081231"
STR_	Store	String	"1001 "
SKU_	SKU	String	"123456 "
sr0_minstk_i	Store Min Stock	Real	"123.456 "
sr0_maxstk_i	Store Max Stock	Real	"123.456 "
sr0_rplmtd_i	Store Replenishment Method	Integer	"1 "

Formatting Conditions

All DAY values should be prefixed with a "D" (case sensitive).

Example of STR_MINMAX.txt File:

"D200812311001 123456 123.456 123.456 1"

Example of STR_MINMAX.csv File:

"D20081231,1001,123456,123.456,123.456,1"

STR_DYNAMIC

Table 4–15 STR_DYNAMIC Data Element Details

Data Type	Data Element Name	Data Description
Measure	Store Service Level	Measure at the SKU/Store/Day level. Used in Dynamic and Poisson Repl Method. Specifies the target percentage level for customer service of a given item at a store. For example, if the goal is to carry enough safety stock to maintain a 90% customer service level, then enter **.90**.
Measure	Store Inventory Selling Days	Measure at the SKU/Store/Day level. Used to specify number of DAYS of forecasted quantity to be used when calculating the Receive Up To Level (RUTL). Used in Hybrid, Dynamic, Poisson, and Loaded SS Dynamic Replenishment Methods. For example, to ensure an order will contain at least 14 days of supply, enter **14**. **Note**: The greater of Inventory Selling Days and Review Time will be used in the RUTL calculation.
Measure	Store Repl Method	Measure at the SKU/Store/Day level. The Replenishment Method determines the type of calculations that will be used in generating a store replenishment plan. The default is No Replenishment.

Table 4–16 STR_DYNAMIC Extracting Program Details

Program Language	Korn Shell Script
Program Name	ro_export_to_aip.sh
Files Used	exportDynamic.xml, createFile.xml and transferFilePositions.xml
Program Type	On request

Table 4–17 STR_DYNAMIC Data Source and Target Details

Data Source Details			Target Data Details	
Data Origin System	RO		Target Object Type	RPAS Measures
Source Object Type	Fixed Length Text File or Comma Separated Values File		Target Object Names	sr0_svclvl_i, sr0_isd_i and sr0_rplmtd_i
Source Object Name	STR_DYNAMIC		Target Object Databases	data/sr0_svclvl_i, data/sr0_isd_i and data/sr0_rplmtd_i
Required/Optional	Optional		Target Object Load Intersection	day_str_sku_

Table 4–18 STR_DYNAMIC Field Level Mapping - Source

Source Fields	Source Field Description	Field Start Position	Field Width
aipefctvdate_	Effective date	1	9
STR_	Store	10	20
ITEM	SKU	30	20
aipreplpval1_	Replenishment Parameter 1	50	10
aipreplpval2_	Replenishment Parameter 2	60	10
aiprplmtd_	Replenishment Method	80	10

Table 4–19 STR_DYNAMIC Field Level Mapping - Target

Target Data Field Name	Target Field Description	Target Field Data Type	Condition/Format
DAY_	DAY	String	"D20081231"
STR_	Store	String	"1001 "
SKU_	SKU	String	"123456 "
sr0_svclvl_i	Store Service Level	Real	"123.456 "
sr0_isd_i	Store Inventory Selling Days	Real	"123.456 "
sr0_rplmtd_i	Store Replenishment Method	Integer	"1 "

Formatting Conditions

All DAY values should be prefixed with a "D" (case sensitive).

Example of STR_DYNAMIC.txt File:

"D200812311001 123456 123.456 123.456 1"

Example of STR_DYNAMIC.csv File:

"D20081231,1001,123456,123.456,123.456,1"

STR_TIMESUPPLY

Table 4–20 STR_TIMESUPPLY Data Element Details

Data Type	Data Element Name	Data Description
Measure	Store Min Time Supply Days	Measure at the SKU/Store/Day level. Used in Time Supply Replenishment Method. It represents minimum stock level in DAYS that are desired on hand to satisfy demand. Min Time Supply Days is used in both safety stock and receipt point calculations. For example, to ensure a minimum inventory level is at least 14 DAYS of supply, enter **14**.
Measure	Store Max Time Supply Days	Measure at the SKU/Store/Day level. Used in the Time Supply Replenishment Method. It represents maximum stock level in DAYS that are desired on hand to satisfy demand. The receipt plan quantity is calculated to bring Net Inventory up to Maximum Time Supply Days. For example, to limit an order to bring the inventory level to no more than to 14 DAYS of supply, enter **14**.
Measure	Store Time Supply Horizon	Measure at the SKU/Store/Day level. The number of DAYS of forecast used to calculate an average forecast (rate of sale). The Rate of Sale (ROS) is then multiplied by the minimum time supply days to generate the safety stock. The Time Supply Horizon can be used to smooth spiky forecasts over a longer time period or extend forecasts if the forecasts are not generated throughout the desired days of supply. Used in Time Supply Replenishment Method. **Note**: When time supply horizon is specified, ROS is used instead of the actual forecast.
Measure	Store Repl Method	Measure at the SKU/Store/Day level. The Replenishment Method determines the type of calculations that will be used in generating a store replenishment plan. The default is No Replenishment.

Table 4–21 STR_TIMESUPPLY Extracting Program Details

Program Language	Korn Shell Script
Program Name	ro_export_to_aip.sh
Files Used	exportDynamic.xml, createFile.xml and transferFilePositions.xml
Program Type	On request

Table 4–22 STR_TIMESUPPLY Data Source and Target Details

Data Source Details			Target Data Details	
Data Origin System	RO		Target Object Type	RPAS Measures
Source Object Type	Fixed Length Text File or Comma Separated Values File		Target Object Names	sr0_mints_day_i, sr0_maxts_day_i, sr0_ts_hzn_i and sr0_rplmtd_i
Source Object Name	STR_TIMESUPPLY		Target Object Databases	data/sr0_mints_day_i, data/sr0_maxts_day_i, data/sr0_ts_hzn_i and data/sr0_rplmtd_i

Table 4–22 (Cont.) STR_TIMESUPPLY Data Source and Target Details

Data Source Details			Target Data Details	
Required/Optional	Optional		Target Object Load Intersection	day_str_sku_

Table 4–23 STR_TIMESUPPLY Field Level Mapping - Source

Source Fields	Source Field Description	Field Start Position	Field Width
aipefctvdate_	Effective date	1	9
STR_	Store	10	20
ITEM	SKU	30	20
aipreplpval1_	Replenishment Parameter 1	50	10
aipreplpval2_	Replenishment Parameter 2	60	10
aipreplAxP1_	Auxiliary Replenishment Parameter 1	70	10
aiprplmtd_	Replenishment Method	80	10

Table 4–24 STR_TIMESUPPLY Field Level Mapping - Target

Target Data Field Name	Target Field Description	Target Field Data Type	Condition/Format
DAY_	DAY	String	"D20081231"
STR_	Store	String	"1001 "
SKU_	SKU	String	"123456 "
sr0_mints_day_i	Store Min Time Supply Days	Real	"123.456 "
sr0_maxts_day_i	Store Max Time Supply Days	Real	"123.456 "
sr0_ts_hzn_i	Store Time Supply Horizon	Real	"123.456 "
sr0_rplmtd_i	Store Replenishment Method	Integer	"1 "

Formatting Conditions

All DAY values should be prefixed with a "D" (case sensitive).

Example of STR_TIMESUPPLY.txt File:

"D200812311001 123456 123.456 123.456 123.456 1"

Example of STR_TIMESUPPLY.csv File:

"D20081231,1001,123456,123.456,123.456,123.456,1"

STR_HYBRID

Table 4–25 STR_HYBRID Data Element Details

Data Type	Data Element Name	Data Description
Measure	Store Min Time Supply Days	Measure at the SKU/Store/Day level. Used in Time Supply Replenishment Method. It represents minimum stock level in DAYS that are desired on hand to satisfy demand. Min Time Supply Days is used in both safety stock and receipt point calculations. For example, to ensure a minimum inventory level is at least 14 DAYS of supply, enter **14**.
Measure	Store Inventory Selling Days	Measure at the SKU/Store/Day level. Used to specify number of DAYS of forecasted quantity to be used when calculating the Receive Up To Level (RUTL). Used in Hybrid, Dynamic, Poisson, and Loaded SS Dynamic Replenishment Methods. For example, to ensure an order will contain at least 14 days of supply, enter 14. **Note**: The greater of Inventory Selling Days and Review Time will be used in the RUTL calculation.
Measure	Store Repl Method	Measure at the SKU/Store/Day level. The Replenishment Method determines the type of calculations that will be used in generating a store replenishment plan. The default is No Replenishment.

Table 4–26 STR_HYBRID Extracting Program Details

Program Language	Korn Shell Script
Program Name	ro_export_to_aip.sh
Files Used	exportHybrid.xml, createFile.xml, and transferFilePositions.xml
Program Type	On request

Table 4–27 STR_HYBRID Data Source and Target Details

Data Source Details			Target Data Details	
Data Origin System	RO		Target Object Type	RPAS Measures
Source Object Type	Fixed Length Text File or Comma Separated Values File		Target Object Names	sr0_mints_day_i, sr0_isd_i and sr0_ rplmtd_i
Source Object Name	STR_HYBRID		Target Object Databases	data/sr0_mints_ day_i, data/sr0_isd_ i and data/sr0_ rplmtd_i
Required/Optional	Optional		Target Object Load Intersection	day_str_sku_

Table 4–28 STR_HYBRID Field Level Mapping - Source

Source Fields	Source Field Description	Field Start Position	Field Width
aipefctvdate_	Effective date	1	9
STR_	Store	10	20
ITEM	SKU	30	20
aipreplpval1_	Replenishment Parameter 1	50	10
aipreplpval2_	Replenishment Parameter 2	60	10
aiprplmtd_	Replenishment Method	80	10

Table 4–29 STR_HYBRID Field Level Mapping - Target

Target Data Field Name	Target Field Description	Target Field Data Type	Condition/Format
DAY_	DAY	String	"D20081231"
STR_	Store	String	"1001 "
SKU_	SKU	String	"123456 "
sr0_mints_day_i	Store Min Time Supply Days	Real	"123.456 "
sr0_isd_i	Store Inventory Selling Days	Real	"123.456 "
sr0_rplmtd_i	Store Replenishment Method	Integer	"1 "

Formatting Conditions

All DAY values should be prefixed with a "D" (case sensitive).

Example of STR_HYBRID.txt File:

"D200812311001 123456 123.456 123.456 1"

Example of STR_HYBRID.csv File:

"D20081231,1001,123456,123.456,123.456,1"

STR_POISSON

Table 4–30 STR_POISSON Data Element Details

Data Type	Data Element Name	Data Description
Measure	Store Service Level	Measure at the SKU/Store/Day level. Used in Dynamic and Poisson Repl Method. Specifies the target percentage level for customer service of a given item at a store. For example, if the goal is to carry enough safety stock to maintain a 90% customer service level, enter **.90**.
Measure	Store Inventory Selling Days	Measure at the SKU/Store/Day level. Used to specify number of DAYS of forecasted quantity to be used when calculating the Receive Up To Level (RUTL). Used in Hybrid, Dynamic, Poisson, and Loaded SS Dynamic Replenishment Methods. For example, to ensure an order will contain at least 14 days of supply, enter **14**. **Note**: The greater of Inventory Selling Days and Review Time will be used in the RUTL calculation.
Measure	Store Repl Method	Measure at the SKU/Store/Day level. The Replenishment Method determines the type of calculations that will be used in generating a store replenishment plan. The default is No Replenishment.

Table 4–31 STR_POISSON Extracting Program Details

Program Language	Korn Shell Script
Program Name	ro_export_to_aip.sh
Files Used	exportPoisson.xml, createFile.xml and transferFilePositions.xml
Program Type	On request

Table 4–32 STR_POISSON Data Source and Target Details

Data Source Details			Target Data Details	
Data Origin System	RO		Target Object Type	RPAS Measures
Source Object Type	Fixed Length Text File or Comma Separated Values File		Target Object Names	sr0_svclvl_i, sr0_isd_i and sr0_rplmtd_i
Source Object Name	STR_POISSON		Target Object Databases	data/sr0_svclvl_i, data/sr0_isd_i and data/sr0_rplmtd_i
Required/Optional	Optional		Target Object Load Intersection	day_str_sku_

Table 4–33 STR_POISSON Field Level Mapping - Source

Source Fields	Source Field Description	Field Start Position	Field Width
aipefctvdate_	Effective date	1	9
STR_	Store	10	20
ITEM	SKU	30	20
aipreplpval1_	Replenishment Parameter 1	50	10
aipreplpval2_	Replenishment Parameter 2	60	10
aiprplmtd_	Replenishment Method	80	10

Table 4–34 STR_POISSON Field Level Mapping - Target

Target Data Field Name	Target Field Description	Target Field Data Type	Condition/Format
DAY_	DAY	String	"D20081231"
STR_	Store	String	"1001 "
SKU_	SKU	String	"123456 "
sr0_svclvl_i	Store Service Level	Real	"123.456 "
sr0_isd_i	Store Inventory Selling Days	Real	"123.456 "
sr0_rplmtd_i	Store Replenishment Method	Integer	"1 "

Formatting Conditions

All DAY values should be prefixed with a "D" (case sensitive).

Example of STR_POISSON.txt File:

"D200812311001 123456 123.456 123.456 1"

Example of STR_POISSON.csv File

"D20081231,1001,123456,123.456,123.456,1"

STR_MINSS

Table 4–35 STR_MINSS Data Element Details

Data Type	Data Element Name	Data Description
Measure	Store Min Safety Stock Units	Measure at the SKU/Store/Day level. Used in Minimum Safety Stock Boundaries calculation to specify the minimum number of UNITS to be kept on hand as safety stock. Used in Hybrid, Poisson, and Dynamic Replenishment Methods. For example, to ensure the number of UNITS of safety stock is at least 10 UNITS, enter **10**. **Note**: If both Min SS DAYS and Min SS UNITS are entered, the larger of min SS units and demand over min SS days is used.

Table 4–36 STR_MINSS Extracting Program Details

Program Language	Korn Shell Script
Program Name	ro_export_to_aip.sh
Files Used	exportMinSafetyStock.xml, createFile.xml and transferFilePositions.xml
Program Type	On request

Table 4–37 STR_MINSS Data Source and Target Details

Data Source Details			Target Data Details	
Data Origin System	RO		Target Object Type	RPAS Measures
Source Object Type	Fixed Length Text File or Comma Separated Values File		Target Object Names	sr0_minss_unt_i
Source Object Name	STR_MINSS		Target Object Databases	data/minss_unt
Required/Optional	Optional		Target Object Load Intersection	day_str_sku_

Table 4–38 STR_MINSS Field Level Mapping - Source

Source Fields	Source Field Description	Field Start Position	Field Width
aipefctvdate_	Effective date	1	9
STR_	Store	10	20
ITEM	SKU	30	20
aipreplaxp2_	Auxiliary Replenishment Parameter 2	80	10

Table 4–39 STR_MINSS Field Level Mapping - Target

Target Data Field Name	Target Field Description	Target Field Data Type	Condition/Format
DAY_	DAY	String	"D20081231"
STR_	Store	String	"1001 "

Table 4–39 (Cont.) STR_MINSS Field Level Mapping - Target

Target Data Field Name	Target Field Description	Target Field Data Type	Condition/Format
SKU_	SKU	String	"123456 "
sr0_minss_unt_i	Store Min Safety Stock Units	Real	"123.456 "

Formatting Conditions

All DAY values should be prefixed with a "D" (case sensitive).

Example of STR_MINSS.txt File:

"D200812311001 123456 123.456"

Example of STR_MINSS.csv File

"D20081231,1001,123456,123.456"

STR_ROUpdate

Table 4–40 STR_ROUpdate Data Element Details

Data Type	Data Element Name	Data Description
Measure	Replenishment Optimization Update for Stores	Mask measure provided by the RO system. This measure is used to limit what data will be extracted out of AIP for the purposes of updating values in RO. The mask also indicates which SKU/stores will be optimized in the next run of RO.

Table 4–41 STR_ROUpdate Extracting Program Details

Program Language	Korn Shell Script
Program Name	ro_export_to_aip.sh
Files Used	exportRoUpdate.xml, createFile.xml and transferFilePositions.xml
Program Type	On request

Table 4–42 STR_ROUpdate Data Source and Target Details

Data Source Details			Target Data Details	
Data Origin System	RO		Target Object Type	RPAS Measures
Source Object Type	Fixed Length Text File or Comma Separated Values File		Target Object Names	iproupdtstri
Source Object Name	STR_ROUpdate		Target Object Databases	data/ro
Required/Optional	Optional		Target Object Load Intersection	str_sku_

Table 4–43 STR_ROUpdate Field Level Mapping - Source

Source Fields	Source Field Description	Field Start Position	Field Width
STR_	Store	1	20
ITEM	SKU	21	20
aipintxupd_	AIP Intersection Update	41	10

Table 4–44 STR_ROUpdate Field Level Mapping - Target

Target Data Field Name	Target Field Description	Target Field Data Type	Condition/Format
STR_	Store	String	"1001 "
SKU_	SKU	String	"123456 "
iproupdtstri	Replenishment Optimization Update for Stores	Boolean	"1 "

Formatting Conditions

None.

Example of STR_ROUpdate.txt File:

"1001 123456 1"

Example of STR_ROUdate.csv File

"1001,123456,1"

Output Measure Data

The table below displays the measure files AIP creates for RO.

Table 4–45 STR_ROUpdate Field Output Measure Data

File Name	Measures Loaded
STR_AIP_DATA	gleadtime_ greview_time_ gpacksize_

STR_AIP_DATA

Table 4–46 STR_AIP_DATA Data Element Details

Data Type	Data Element Name	Data Description
Measure	Lead Time	GLEADTIME_
Measure	Review Time	GREVIEW_TIME_
Measure	Pack Size	GPACKSIZE_

Table 4–47 STR_AIP_DATA Extracting Program Details

Program Language	Korn Shell Script
Program Name	aip_export_to_ro.sh
Files Used	N/A
Program Type	On request

Table 4–48 STR_AIP_DATA Data Source and Target Details

Data Source Details			Target Data Details	
Data Origin System	AIP		Target Object Type	RPAS Measures
Source Object Type	Fixed Length Text File or Comma Separated Values File		Target Object Names	gleadtime_, greview_time_ and gpacksize_
Source Object Name	STR_AIP_DATA		Target Object Databases	data/calcs
Required/Optional	Optional		Target Object Load Intersection	str_item

Table 4–49 STR_AIP_DATA Field Level Mapping - Source

Source Fields	Source Field Description	Field Start Position	Field Width
STR_	Store	1	20
SKU_	SKU	21	20
iplti	Lead Time	41	10
iprtv	Review Time	51	10
dmx_pszmap	Pack Size Map	61	10

Table 4–50 STR_AIP_DATA Field Level Mapping - Target

Target Data Field Name	Target Field Description	Target Field Data Type	Condition/Format
STR_	Store	String	"1001 "
ITEM	SKU	String	"123456 "
gleadtime_	GLEADTIME_	Real	"123.456 "
greview_time_	GREVIEW_TIME_	Real	"123.456 "
gpacksize	GPACKSIZE_	Real	"123.456 "

Formatting Conditions

None.

Example of STR_AIP_DATA.txt File:

"1001 123456 123.456 123.456 123.456"

Example of STR_AIP_DATA.csv File

"1001,123456,123.456,123.456,123.456"

Batch Logs and Settings

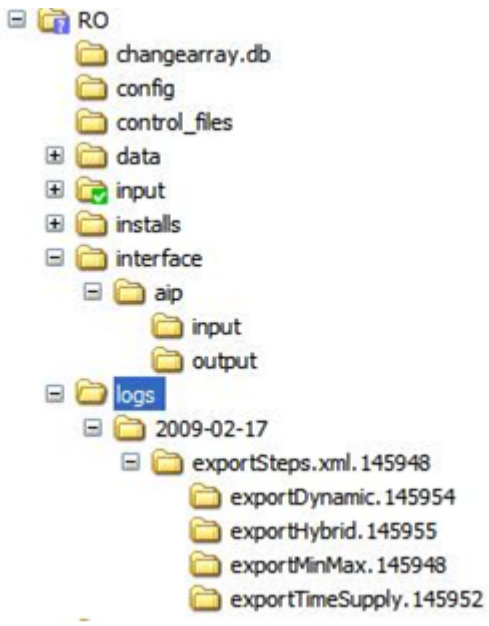
The environment variable RO_LOG_LEVEL can be set to one of the following log levels:

- PROFILE
- DEBUG
- INFORMATION
- WARNING
- ERROR

The default is ERROR.

The logs are created in the directory specified in environmental variable RO_LOG_HOME. The logs will be contained in a directory with today's date.

Figure 4–4 RO Logs



5

Configuration Considerations

This chapter provides information on the configuration changes that can be made for RO. For some retailers, parts of the released version of the RO configuration may fit perfectly. However, it is anticipated that changes are needed to make the RO configuration match the organization of the retailer.

Hierarchies are limited to the determination of hierarchy aspects that pertain directly to dimensions, attributes, and facts. Due to RPAS limitations on intersection, distinct hierarchies must exist for the construction of all intersections to support all facts. No more than one dimension from any hierarchy may exist in a measure intersection.

For information on the configuration changes that can be made, see the following sections:

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

Calendar (CLND) Hierarchy

The calendar hierarchy represents time in all RPAS solutions. It is a required hierarchy and must have the DAY dimension. As it relates to RO, the calendar hierarchy is needed to store time-phased measures.

Your implementation can structure the calendar hierarchy in any way that best suits your fuctional needs. Dimensions other than DAY have been included in RO for the purpose of illustration. They can be modified or removed without requiring changes to any other elements of the configuration. Other dimensions and hierarchy branches can also be added without requiring changes to other elements of the configuration.

Product (PROD) Hierarchy

The product hierarchy (also known as the merchandise hierarchy) represents the retailer's merchandise, that is, merchandise that the retailer retails through its retail channels. RO does not enforce any constraints on the structuring of this hierarchy, but the configuration does use a few levels of this hierarchy extensively in workbook wizards, labeled intersections, rules, position queries, and measure values (Single Hier Select measures). Any changes to this hierarchy must be accompanied by changes to all these elements if they employ the particular level that is being modified or removed. Adding levels or branches or changing labels should not require any changes to the configuration.

From the product hierarchy, the configuration employs ITEM and at least one more dimension higher than ITEM. You may choose the higher level dimension. To reduce patching and upgrade efforts, it is recommended that the names of these levels be left unchanged unless absolutely necessary.

ITEM is necessary because it is the level at which the replenishment recommendations are made. The higher level dimension that you choose is necessary because the optimization goals and optimization constraints are made at this level.

Location (LOC) Hierarchy

The location hierarchy represents the retailer's retail locations and their roll-ups. RO imposes a few constraints on the structure of this hierarchy, but for the most part it is flexible.

From the location hierarchy, the configuration employs the store (STR) dimension and at least one more dimension higher than STR, such as region (RGN).

STR is necessary because it is the level at which the replenishment recommendations are made. The higher level dimension that you choose is necessary because the optimization goals and optimization constraints are made at this higher level.

6

Batch Processing

This chapter contains all the scripts that are needed to execute and maintain the RO environment.

Data is loaded into RO using the standard RPAS approach. See the *Oracle Retail Predictive Application Server Administration Guide* for details on formatting the load data files and on the utilities that enable administrators to load data into RPAS.

Note: Comma-separated values (CSV) files are recommended to reduce the sizes of load files.

Batch Script Summary

Table 6–1 describes the directories that are used by the batch scripts. These directories are subdirectories of the `<ro_directory>` directory. The `<ro_directory>` is defined by the implementer.

Table 6–1 Directories Used by Batch Scripts

Directory Name	Content of the Directory
bin	Batch scripts
config	RO template configuration
domain	Domains
input	Input files for building the domain
logs	Log files from running any of the batch scripts
temp	Temporary files used by the batch scripts

Batch Script Summary Table

Table 6–2 summarizes the available batch scripts. The batch scripts are located in the `<ro_directory>/bin` directory.

The following information is included in the tables for each batch script:

- A short description of the script
- The name of the script
- Dependencies on other batch scripts

Table 6–2 Batch Script Summary

Script Name	Description	Suggested Frequency	Dependencies
ro_optbatch.ksh	Optimization Batch: Master Domain	Periodically	None
ro_optbatch_ localdomain.sh	Optimization Batch: Local Domain	Periodically	None
ro_replbatch.sh	Replenishment Batch: Master Domain	Periodically	None
ro_replbatch_ localdomain.sh	Replenishment Batch: Local Domain	Periodically	None
Backup	Backup	Daily	None

There are two ways to check if a batch completed successfully:

- In the batch log file, check for any errors, exceptions, or failures. If there are none, the batch completed successfully.
- A generation ID is used when a batch runs successfully. If a generation ID is available in the wizard process for the approve workbook, this indicates that the batch ran successfully.

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

Batch Designs

This section contains detailed information on the following batch scripts:

- Optimization Batch Script: Master Domain
- Optimization Batch: Local Domain
- Replenishment Batch: Master Domain
- Replenishment Batch: Local Domain

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

Optimization Batch Script: Master Domain

The optimization batch has two modes: full mode and refresh mode. The optimization batch full mode is run only periodically, either after new simulation results from APC RO are provided or after a dramatic, widespread change in the sales pattern in the domain. The optimization batch generates optimal and constrained service levels versus inventory cost curves when running in full mode. A user can select the targeted service level and decide the system-recommended replenishment settings per item/store combination by using the Optimization Review workbook after a full mode optimization batch is run.

ro_optbatch.ksh should run from a master domain. It will loop over all local domains.

Script
ro_optbatch.sh

Usage
ro_optbatch.sh [-d {masterpath}] [-noparallel | -maxprocesses {n}]

Argument	Description
-d {masterpath}	The master domain path.
-noparallel	Specifies that no-parallel processing should be used.
-maxprocesses	Specifies the maximum number of processes used in the optimization batch.

Optimization Batch: Local Domain

The optimization batch for local domains also has two modes: full mode and refresh mode. The refresh mode batch should be run after every data load, except when the full mode optimization batch is run. The refresh mode batch updates the system-recommended replenishment settings based on new sales history and previously approved full mode batch result without any user involvement.

ro_optbatch_localdomain.ksh is useful for running the batch only in a local domain.

Script
ro_optbatch_localdomain.ksh

Usage
ro_optbatch_localdomain.sh [-d {localpath}]

Argument	Description
-d {localpath}	The local domain path.

Replenishment Batch: Master Domain

The replenishment batch generates projected inventory, service levels, and lost sales for the next 13 weeks using the system-recommended replenishment parameters and current replenishment parameters. It runs after each optimization batch run (full mode or refresh mode). The results of replenishment batch can be reviewed in the Replenishment Analyst workbook.

ro_replbatch.ksh should run from a master domain. It will loop over all local domains.

Script
ro_replbatch.ksh

Usage
ro_replbatch.sh [-d {masterpath}] [-noparallel | -maxprocesses {n}]

Argument	Description
-d {masterpath}	The master domain path.
-noparallel	Specifies that no-parallel processing should be used.
-maxprocesses	Specifies the maximum number of processes used in the optimization batch.

Replenishment Batch: Local Domain

The replenishment batch generates projected inventory, service levels, and lost sales for the next 13 weeks using the system-recommended replenishment parameters and current replenishment parameters. It runs after each optimization batch run (full mode or refresh mode). The results of replenishment batch can be reviewed in the Replenishment Analyst workbook.

ro_replbatch_localdomain.ksh is used for running the batch from only a local domain.

Script
ro_replbatch_localdomain.ksh

Usage
ro_replbatch_localdomain.sh [-d {localpath}]

Argument	Description
-d {localpath}	The local domain path.

7

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. RO has been internationalized to support multiple languages.

This section describes configuration settings and features of the software that ensure that the base application can handle multiple languages.

Translation

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

- Graphical user interface (GUI)
- Error messages

The following components are not usually translated:

- Documentation (Online Help, Release Notes, Installation Guide, User Guide, Operations Guide)
- Batch programs and messages
- Log files
- Configuration Tools
- Reports
- Demo data
- Training Materials

The user interface for RO has been translated into the following languages:

- German
- French
- Spanish
- Japanese
- Traditional Chinese
- Simplified Chinese
- Korean
- Brazilian Portuguese

- Russian
- Italian

Note: For more information about internationalization, see the *RPAS Administration Guide*.