Oracle® Retail Advanced Inventory Planning

Operations Guide

Release 15.0.3

F12198-03

January 2020

ORACLE®

Oracle Retail Advanced Inventory Planning Operations Guide, Release 15.0.3

F12198-03

Copyright © 2020, Oracle and/or its affiliates. All rights reserved.

Primary Author: Melissa Artley

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

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

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

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

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

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

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

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

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

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

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

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, “alteration” refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all

reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

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

Contents

4 AIP RPAS Batch Processing

5 AIP RPAS Daily Batch Scripts

6 AIP Java/Oracle Batch Process Flow

7 AIP Java/Oracle Daily Batch Process Details

8 AIP Interval Batch Scripts

9 AIP Batch Environment Maintenance

10 AIP Daily Process

11 AIP RPAS Intra-day Batch Processing

12 AIP RPAS Intra-day Batch Scripts

List of Figures

List of Tables

Send Us Your Comments

Oracle Retail Advanced Inventory Planning Operations Guide, Release 15.0.3.

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

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

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

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

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

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

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

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

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

Preface

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

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

Audience

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

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

Documentation Accessibility

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

Access to Oracle Support

Oracle customers 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 Advanced Inventory Planning Release 15.0.3 documentation set:

- *Oracle Retail Advanced Inventory Planning Implementation Guide*
- *Oracle Retail Advanced Inventory Planning Installation Guide*
- *Oracle Retail Advanced Inventory Planning Operations Guide*
- *Oracle Retail Advanced Inventory Planning Order Management User Guide*
- *Oracle Retail Advanced Inventory Planning Release Notes*
- *Oracle Retail Advanced Inventory Planning Store and Warehouse Replenishment Planning User Guide for the RPAS Fusion Client*

The following documentation may also be needed when implementing AIP:

- Oracle Retail Planning Batch Script Architecture (BSA) Implementation Guide
- Oracle Retail Integration Bus (RIB) documentation, based on type of deployment
- Oracle Retail Extract Transform and Load (RETL) documentation
- Oracle Retail Predictive Application Server (RPAS) documentation

My Oracle Support Documents

These Oracle Retail Advanced Inventory Planning Release 15.0.x documents are available on My Oracle Support:

- *Oracle Retail Advanced Inventory Planning Calculations for Store and Warehouse Replenishment Planning* (Doc ID 2075628.1)
- *Oracle Retail Advanced Inventory Planning Order Review and Approval* (Doc ID 2076972.1)
- *Oracle Retail Advanced Inventory Planning Supply Chain Creation 15.x* (Doc ID 2081101.1)
- *Oracle Retail Advanced Inventory Planning XREF 15.0.3* (Doc ID 2021275.1)

Customer Support

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

https://support.oracle.com

When contacting Customer Support, please provide the following:

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

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 15.0) or a later patch release (for example, 15.0.3). If you are installing the base release, additional patch, and bundled hot fix releases, read the documentation

for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

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

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

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

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

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

Oracle Retail Documentation on the Oracle Technology Network

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

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

(Data Model documents are not available through Oracle Technology Network. You can obtain them through My Oracle Support.)

Conventions

The following text conventions are used in this document:

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

1

About Advanced Inventory Planning

This chapter provides general information on these topics:

- AIP Architecture
- RPAS Platform
- AIP Java/Oracle Platform

> **Note:** Information about Batch Script Architecture (BSA) previously found in this guide is now in the *Oracle Retail Planning Batch Script Architecture (BSA) Implementation Guide*.

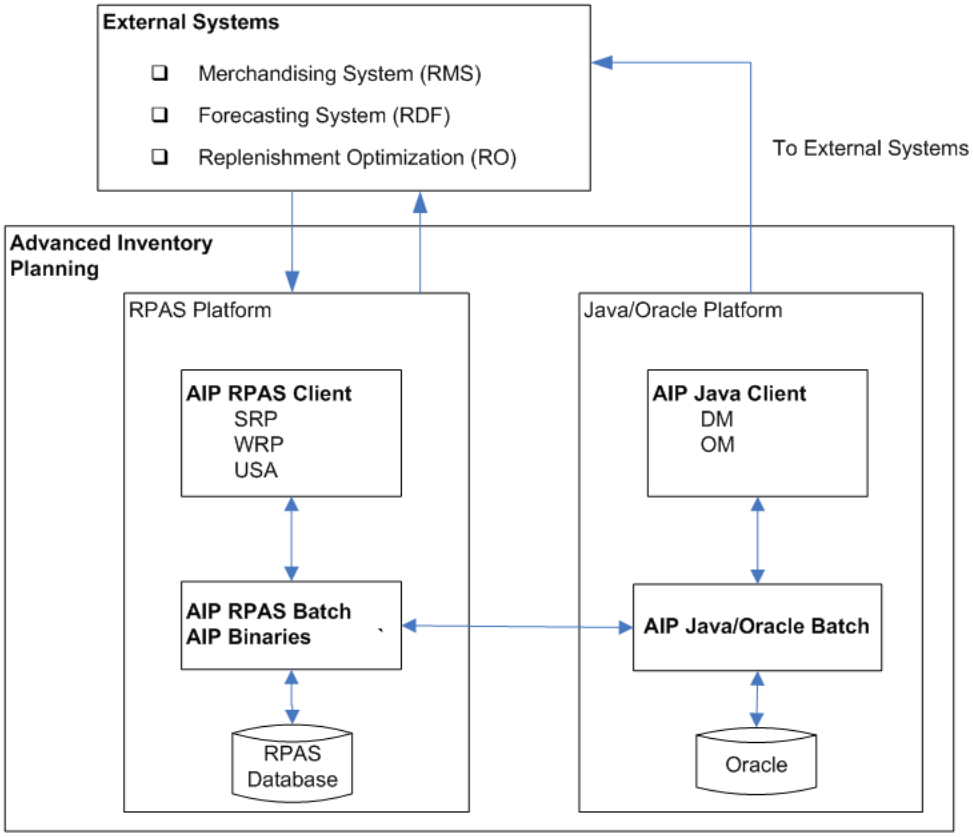
AIP Architecture

The AIP architecture consists of the AIP modules distributed across these two platforms:

- RPAS platform
- Java/Oracle platform

The external merchandising system, the forecasting system, and the replenishment optimization system are integrated with AIP to provide the inventory/foundation data and the forecasting data to AIP to effectively plan the inventory flow across the retailers supply chain. AIP can integrate with any merchandising or forecasting systems. Figure 1–1 shows the integrated AIP solution with the Oracle Retail Merchandising System (RMS) and the Oracle Retail Demand Forecasting (RDF) system and Oracle Retail Replenishment Optimization (RO) system.

Figure 1–1 AIP Architecture



Acronym	Definition
RMS	Oracle Retail Merchandising System
RDF	Oracle Retail Demand Forecasting
RO	Oracle Retail Replenishment Optimization
OM	Order Management
RPAS	Retail Predictive Application Server
DM	Data Management
OM	Order Management
SRP	Store Replenishment Planning
WRP	Warehouse Replenishment Planning
USA	User Specified Allocation

RPAS Platform

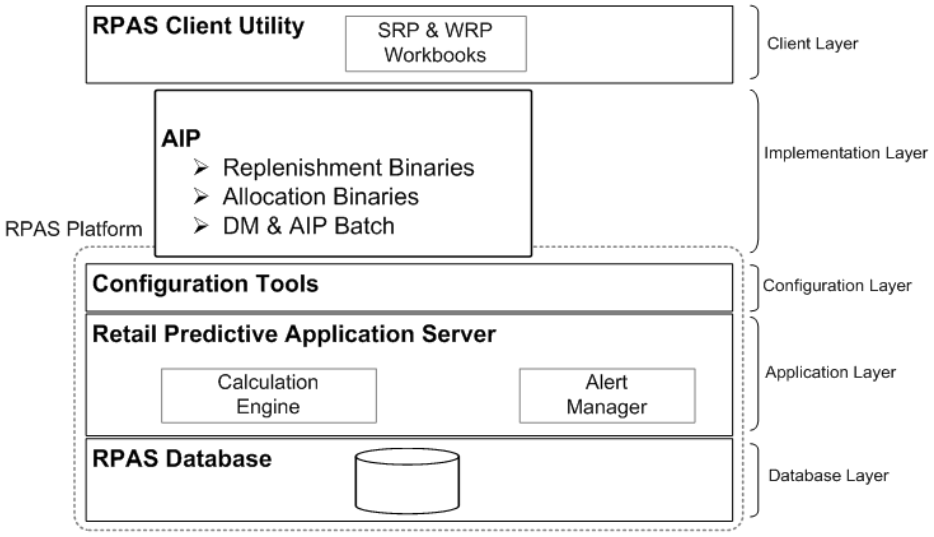
The replenishment and allocation calculations across the supply chain for stores and warehouses are implemented using the C++ component architecture. The calculations are provided as binaries and are integrated using rule groups and then executed using the AIP RPAS batch process.

To modify and update the parameters for store planning, or to manage the alerts raised during the batch process, AIP provides these client utilities:

- Store Replenishment Planning (SRP)
- User Specified Allocation (USA)

Similarly, to modify and update the parameters for warehouse planning, AIP provides the Warehouse Replenishment Planning (WRP) client utility.

Figure 1–2 RPAS Platform



AIP RPAS Platform

The RPAS platform is used for implementing the predictive planning solutions. For AIP, the RPAS platform provides the following layers:

Layer	Description
Client layer	RPAS provides a client utility for users to work on the worksheets within the workbooks. The users view and maintain the measures for the replenishment and allocation calculations across the supply chain. The user interfaces for SRP, WRP, and USA are provided by the client utility. Users need to install the client utility provided by the RPAS platform and configure the SRP and WRP user interfaces. Based on access rights, the AIP planner can log in to the different workbooks.
Implementation layer	The AIP solution is implemented on RPAS by using all the features and functions provided by the RPAS planning platform. AIP is implemented using the C++ component architecture, and is executed in the form of binaries during the AIP batch process.
Configuration layer	RPAS provides the Configuration Tools. The RPAS Configuration Tools are used by the AIP implementers to set up the workbooks, measures, and rules for the business needs of the retailer.
Application layer	The Alert Manager is associated with the workbooks. Alerts based on business scenarios and thresholds are designed in the alert manager. When the planners log in to the system using the client utility, the alert manager window pops up with all the alerts listed. This enables the planners to take appropriate actions.

Layer	Description
Database layer	RPAS provides a multi-dimensional database, for AIP to support the following: ■ Different hierarchies ■ Flexibility to navigate to various intersections, across all the hierarchies ■ Ability to easily modify values across any point in the hierarchy, and apply changes across all the levels by using the spreading and aggregating techniques

AIP Java/Oracle Platform

Note: AIP Java/Oracle, AIP on Oracle, and AIP online are often used interchangeably to refer to those parts of AIP that access the Oracle relational database. This includes the Data Management and Order Management GUI components and a host of UNIX shell scripts and PL/SQL modules.

The two AIP online modules, Data Management and Order Management, are Web-based applications that interact with an Oracle database and run on an application server. Using a Web browser, users can log in to the multi-user graphical user interface, an applet.

Data Management (DM) Module

The Data Management (DM) module is implemented across both the Oracle and RPAS platforms to manage the supply chain parameters across the retailer's stores and warehouses.

The Data Management module has two parts that operate on the Oracle database:

Part	Description
DM Batch	This is an extensive set of processes that involves extracting data for use in RPAS, loading data updated or created in RPAS and also performing a number of steps to automatically setup and maintain the supply chain.
DM Online Interface	This Java applet client utility is used by the AIP planners for managing the supply chain.

The DM Batch processes execute on both the RPAS platform and the Oracle platform. It is used to sync and source data across all the AIP modules.

Note: On the RPAS platform, the DM module has only the batch process. There is no user interface.

Order Management (OM) Module

The Order Management (OM) module is used to manage the purchase orders and transfers generated by the planning calculations, and then release them to RMS. The OM module operates only on the Java/Oracle platform.

The Order Management module has two parts that operate on the Oracle platform:

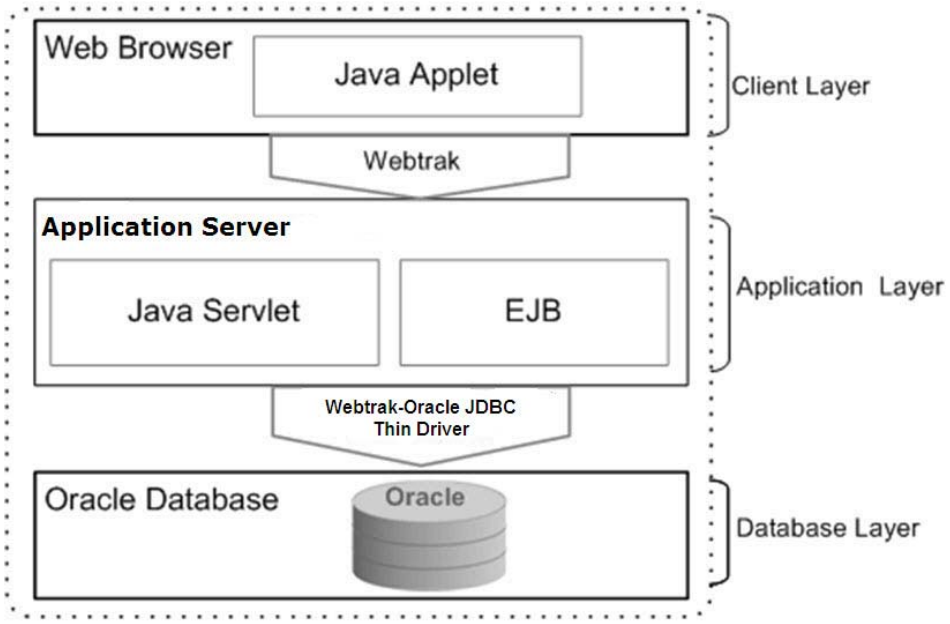
Part	Description
OM Batch	This process consists of loading the planned purchase orders and transfers from RPAS into the Oracle database and then releasing them to RMS on their lead times.
OM Online Interface	This is a Java applet client utility used by AIP for manually creating purchase orders, releasing purchase orders early, and managing the purchase orders and transfers which have been released to RMS.

AIP provides a client utility for the OM module, and it also has a batch process for a daily mass release of purchase orders and transfers which have met their lead time.

AIP Java/Oracle Platform

Figure 1–3 shows the AIP Java/Oracle Platform.

Figure 1–3 AIP Java/Oracle Platform



The AIP Java application is designed with the following layers:

Layer	Description
Client layer	AIP Online provides a Graphical User Interface (GUI), which is a Java AWT applet. Users can log in by using a Web browser. Both the DM and OM modules are accessed through the Web interface. A common user account is created by the AIP Online administrator so that the user can access both the DM and OM modules.
Application layer	The applet communicates with an application server through an Oracle Retail WebTrack communication layer. The server runs a Java servlet, which provides a data access layer.
Database layer	The server communicates through the Oracle Retail WebTrack and an Oracle JDBC driver to an Oracle database.

2

AIP Integration

The AIP solution uses the following Oracle Retail integration tools to integrate within AIP and also with other Oracle Retail products:

- Oracle Retail Integration Bus (RIB)
- Oracle Retail Extract, Transform, and Load (RETL)

AIP is distributed across two platforms: the RPAS and Java/Oracle platforms. There are two types of integration in AIP: external and internal integration.

External Integration

The AIP solution integrates with the Oracle Retail Merchandising System (RMS), Oracle Retail Demand Forecasting (RDF), Oracle Retail Analytic Science (ORASE) and Oracle Retail Replenishment Optimization products.

Figure 2–1 AIP External Integration

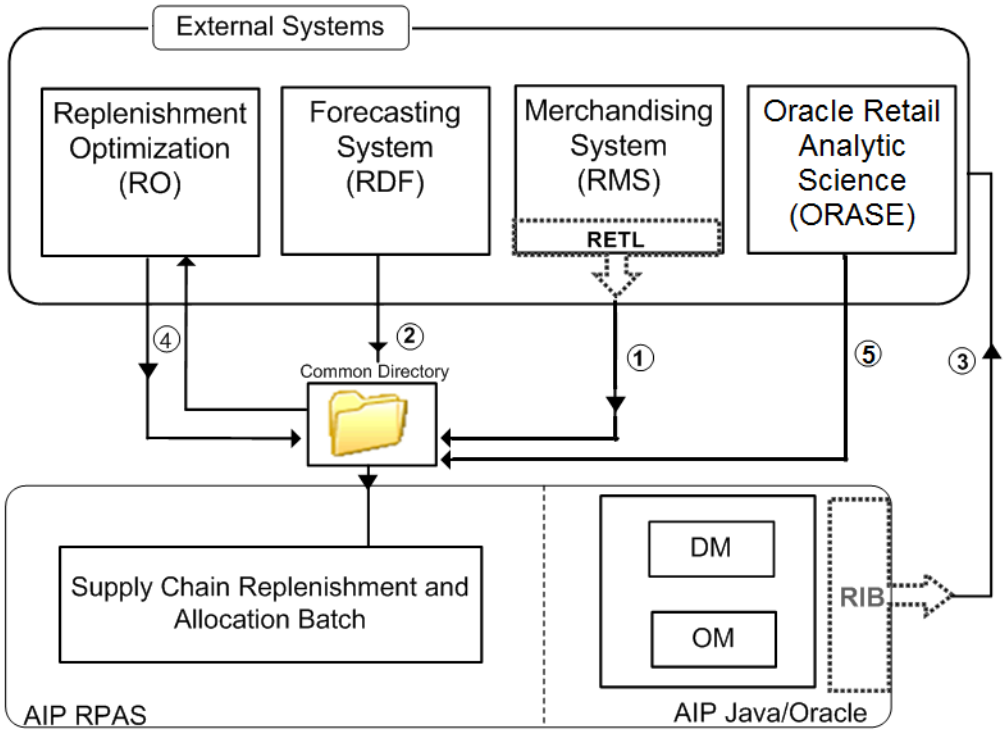


Table 2–1 describes the external integration between AIP and other products as shown in Figure 2–1.

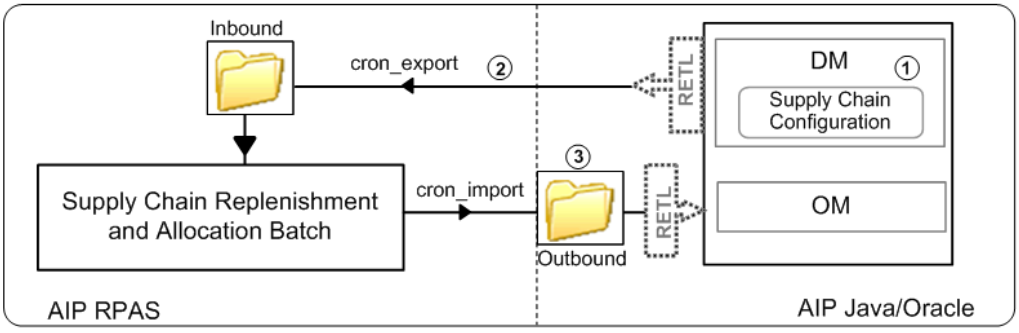
Table 2–1 External Integration Between AIP and Other Products

Phase	Integration between...	Description
1	RMS and AIP using RETL	The data required by AIP from RMS is extracted. The data from RMS Oracle tables are transformed into an AIP compatible format (text files) using the custom script that calls the Oracle Retail Extract, Transform, and Load (RETL). The extracted text files are placed in the common directory for AIP batch to access these files. AIP batch scripts further processes these text files and loads them into the RPAS platform and AIP Online platform. **Note**: Part of the RMS data is loaded into DM Online during the internal integration of AIP.
2	RDF and AIP	The retailer should run the RDF extract scripts to create the forecasting measure data flat files in the format AIP expects. Then the AIP batch scripts load the RDF measures into the RPAS database.
3	AIP and RMS using RIB	After AIP has completed a replenishment plan across the retailer's supply chain the plan is passed to OM. OM determines which purchase orders and transfers have met their lead time and therefore must be passed to RMS for execution. The purchase orders that are manually created in OM Online must also be passed to RMS. All purchase orders and transfers that are to be executed in RMS are passed to RMS using the Oracle Retail Integration Bus (RIB). RIB is a near real-time data synchronization solution used by AIP for publishing orders to RMS. AIP publishes two sets of order messages to RIB: purchase orders and transfers. RMS subscribes to the RIB messages and inserts the orders into the appropriate RMS purchase order and transfer tables.
4	AIP and RO	RO is a replenishment optimization system and the integration of AIP and RO is an optional setup for Retailers. Lead Time, Review Time and Preferred Pack Size data is exported out of AIP in flat files for use when optimizing the replenishment settings. AIP then loads the optimized replenishment parameters from flat files provided by RO. The integration with RO occurs infrequently such as monthly or quarterly. Retailers who do not run RO should exclude the related AIP batch scripts from their job scheduler tasks.
5	AIP and ORASE	The integration of AIP and Oracle Retail Analytic Science Engine (ORASE) is optional set up for Retailers. Demand Transfer Percentage data is exported from ORASE to AIP for use in substitution.

Internal Integration

The process of integrating the AIP Java/Oracle platform with the AIP RPAS platform is called internal integration.

Figure 2–2 AIP Internal Integration



AIP uses RETL integration tool to integrate the AIP modules that are spread across both the platforms. RETL is used because it provides the flexibility to handle large amounts of data flow across the RPAS database and the AIP Online (Oracle) platforms.

Table 2–2 describes the internal integration between AIP components as shown in Figure 2–2.

Table 2–2 Internal Integration Between AIP Components

Phase	Integration	Description
1	Supply Chain Configuration	The AIP Planner uses the DM Online application to initially set up and maintain the supply chain by establishing the connectivity between the stores, warehouses, and suppliers. This is done after the initial batch of AIP is run. The DM Online application is also used for other configurations that are used during the AIP batch execution. For example, the retailer introduces new warehouses and stores in RMS. They need to be configured in the DM Online application to establish the sources of the warehouses and stores as well as the many other configurations that the AIP solution needs to know prior to the AIP replenishment planning calculations. **Note**: The RMS information is passed into DM Online during the AIP Batch execution (cron_import).
2	RETL Extract Scripts	The RETL extract scripts are executed during the batch process to move the DM Online parameters from the AIP Oracle database to an outbound staging directory. The client's job scheduler FTPs or copies the files in the directory to the inbound directory of the AIP RPAS platform.
3	SRP and WRP	The store receipt plan and the warehouse receipt plan are exported out of the AIP RPAS platform for transfer by a client's job scheduling process to the Order Management module in the Java/Oracle platform. Also, RMS data required in DM on the Java/Oracle platform are also exported out of the RPAS platform for transfer to the Oracle database.

Internal Integration Architecture

The internal integration allows the two AIP databases to communicate data that must be shared by both. The interface into the Oracle database uses the following technologies:

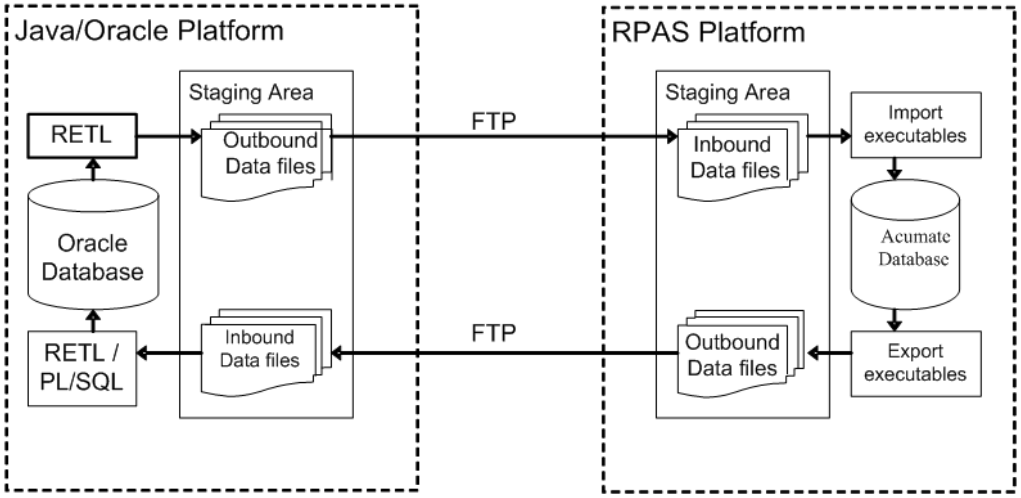
- Oracle Retail Extract, Transform, and Load (RETL) scripts
- Oracle PL/SQL packages
- Shell scripts

The interface to the RPAS database uses the following technologies:

- Shell scripts
- RPAS utilities

As shown in Figure 2–3, the data is communicated between the databases on a nightly basis. Once data is exported the client's scheduler should transfer the outbound files to the receiving platform/directory. Then, another scheduled job in client's scheduler executes the shell scripts that trigger the import processing of the files.

Figure 2–3 AIP Interfaces

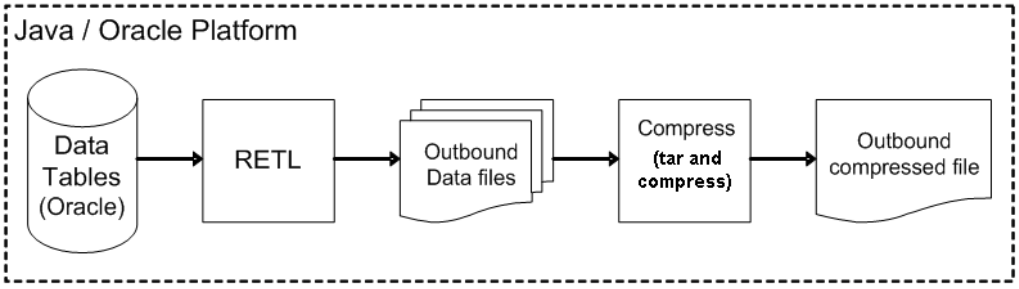


Exporting

The export process involves running the export RETL scripts to create the outbound data files. The process compresses them into two files, and then places the files in a staging area where they can be moved to the RPAS platform, through file copy or FTP, by a job scheduling application. RETL can export the data directly from the Oracle tables, with no additional processing.

Note: When RETL creates outbound data files directly from an Oracle database with multi-byte characters, warning messages are written to the log file regarding a mismatch between input and output field lengths. This is due to a difference in how RETL and Oracle interpret multi-byte characters and does not represent a problem with the export process.

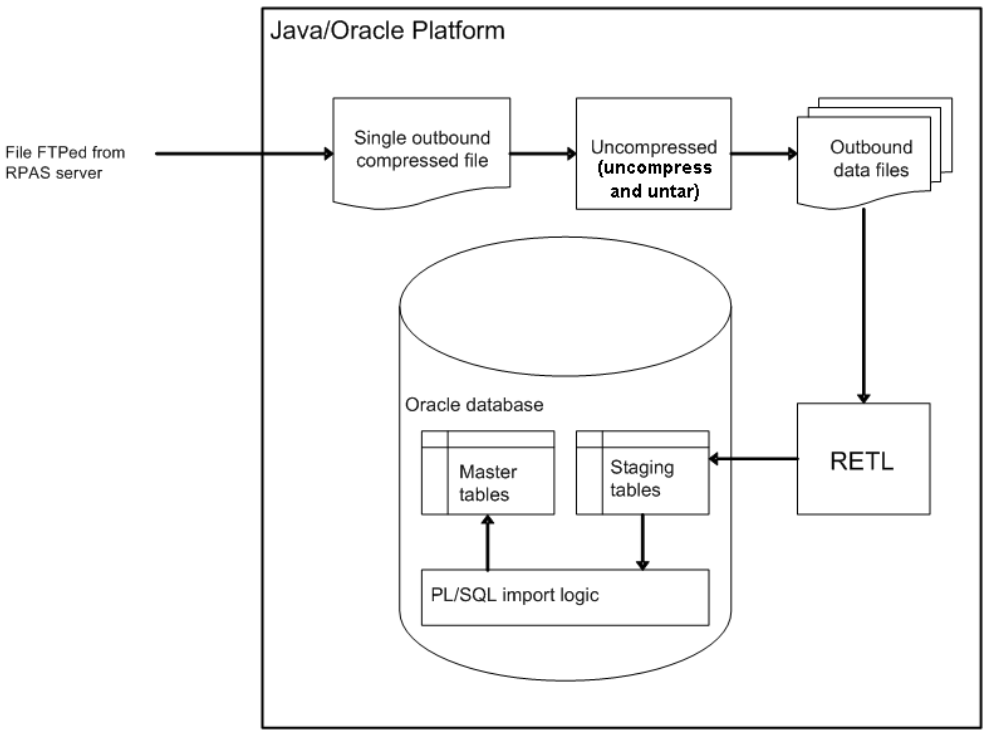
Figure 2–4 Exporting from the Oracle Database



Importing

The import process is slightly more complex than the export process. The data files must be moved through file copy or FTP by a job scheduling application from the RPAS server location. The files are then decompressed before they are processed through RETL. Due to the required additional pre-insert processing, RETL is not able to perform a direct import to the database. For instance, duplicate entries must be checked so that updates are performed, rather than inserted. This introduces another layer of processing in the form of PL/SQL packages embedded in the database. Data is first imported to the staging tables by RETL, and then the PL/SQL logic is executed to update the master tables.

Figure 2–5 Importing into the Oracle Database



RETL Configuration

The interface process is designed to be fully automated once configured, with support from a retailer's custom process for moving the files between the AIP Oracle and AIP RPAS server locations. The config.xml needs to be configured to the specific environment. This file is located under the root integration directory (integration/).

Note: Refer to the *Oracle Retail Advanced Inventory Planning Installation Guide* for installation instructions. Refer to the *Oracle Retail Advanced Inventory Planning Implementation Guide* for configuration details.

3

AIP Interfaces and Transformation Scripts

The AIP system interfaces with the merchandising system, the forecasting system, and any other external system that provides data in the predetermined file format. This chapter provides information on how AIP interfaces with various RMS versions and RDF. For a complete list of supported RMS versions, see the *Oracle Retail Advanced Inventory Planning Installation Guide.*

RMS to AIP Interfaces and Transformation Scripts

During AIP installation, the RMS-AIP transform scripts and output schema files were installed to a directory specified by the user, as the Dir to store RMS transform files. This location is referred to as $RMS_AIP_TRANSFORM for this section. Refer to the *Oracle Retail Advanced Inventory Planning Installation Guide* for more information about the installation details.

The AIP installation includes a separate set of transformation scripts for interfacing with RMS 10, 11, and 13.0/13.1. These were installed to $RMS_AIP_TRANSFORM/k_shell_scripts. The schema files in $RMS_AIP_TRANSFORM/aip_schema_dir are used for interfacing with any version of RMS. The following documentation describes the differences in the interfacing process for each version of RMS.

The overall process of interfacing RMS with AIP begins with generation of the RMS RETL extracts. A separate set of RMS extract scripts and schema files control this generation. These scripts and schema are not included in the AIP installation.

After generating the RMS RETL extracts, the RMS-AIP transformation scripts convert the RMS RETL extracts into RPAS loadable format. There are two sets of schema files: one which matches the formats of the RMS RETL extract files, and a second which matches the formats of the RPAS-loadable files.

Note: If the input file has multibyte characters then the user has to manually change the default value of **LC_ALL** in `aip_constants_rpas.sh` based on their locale in order to process multibyte characters.

AIP programmatically process multibyte character based on the domain-property `fixedwidth_utf8` and the exported value of environment variable **LC_ALL**.

Transform Interface Scripts

Location: $RMS_AIP_TRANSFORM/k_shell_scripts and $RPAS_HOME/bin

The Transform Interface Scripts are the Korn shell scripts used to drive the process for transforming the RMS RETL extract files into AIP RPAS domain-loadable files.

The $RMS_AIP_TRANSFORM/k_shell_scripts directory's contents should have been copied to $RPAS_HOME/bin during installation.

The execution of these scripts is controlled by running the master script:

- aip_t_master_rms10.ksh for RMS 10.x
- aip_t_master_rms11.ksh for RMS 11.x
- aip_t_master_rms13.ksh for RMS 13.1/13.0
- aip_t_master_rms13.2.ksh for RMS 13.2
- aip_t_master_rms14.0.ksh for RMS 14.0

These scripts are run before running the AIP RPAS batch.

RMS 14.0 Transform Interface Scripts

The execution of these scripts is controlled by running the master script, aip_t_master_rms14.0.ksh for RMS 14.0 transforms.

These scripts are run before running the AIP RPAS batch.

For RMS 14.0, $RPAS_HOME/bin must contain the following scripts:

- aip_t_master_rms14.0.ksh
- aipt_clnd.awk, aipt_clnd.ksh, days_from_cycle_start_date.pl
- aipt_delete_input_files.ksh
- aipt_future_delivery_rms14.0.ksh
- aipt_item_rms14.0.ksh
- aipt_move.ksh
- aipt_orghier_rms14.0.ksh
- aipt_rename.ksh
- aipt_sr0_dayslsld_rms14.0.ksh
- aipt_str_prd_life_rms14.0.ksh
- aipt_sub_item_rms14.0.ksh

RMS 13.2 Transform Interface Scripts

The execution of these scripts is controlled by running the master script, aip_t_master_rms13.2.ksh for RMS 13.2 transforms.

These scripts are run before running the AIP RPAS batch.

For RMS 13.2, $RPAS_HOME/bin must contain the following scripts:

- aip_t_master_rms13.2.ksh
- aipt_clnd.awk, aipt_clnd.ksh, days_from_cycle_start_date.pl
- aipt_delete_input_files.ksh
- aipt_future_delivery_rms13.2.ksh
- aipt_item_rms13.2.ksh
- aipt_move.ksh

- aipt_orghier_rms13.2.ksh
- aipt_rename.ksh
- aipt_sr0_dayslsld_rms13.2.ksh
- aipt_str_prd_life_rms13.2.ksh
- aipt_sub_item_rms13.2.ksh

RMS 13.1/13.0 Transform Interface Scripts

The execution of these scripts is controlled by running the master script, aip_t_master_rms13.ksh for RMS 13.1/13.0 transforms.

These scripts are run before running the AIP RPAS batch.

For RMS 13.1/13.0, $RPAS_HOME/bin must contain the following scripts:

- aip_t_master_rms13.ksh
- aipt_clnd.awk, aipt_clnd.ksh, days_from_cycle_start_date.pl
- aipt_delete_input_files.ksh
- aipt_future_delivery_rms13.ksh
- aipt_item_rms13.ksh
- aipt_move.ksh
- aipt_orghier_rms13.ksh
- aipt_pad.ksh, aipt_pad_splr.awk, aipt_pad_whse.awk
- aipt_rename.ksh
- aipt_sr0_dayslsld_rms13.ksh
- aipt_str_prd_life_rms13.ksh
- aipt_sub_item_rms13.ksh

RMS 11 Transform Interface Scripts

The execution of these scripts is controlled by running the master script, aip_t_master_rms11.ksh for RMS 11 transforms. These scripts are run before the AIP RPAS batch is run.

For RMS 11, $RPAS_HOME/bin must contain the following scripts:

- aip_t_master_rms11.ksh
- aipt_clnd.awk, aipt_clnd.ksh, days_from_cycle_start_date.pl
- aipt_delete_input_files.ksh
- aipt_future_delivery_rms11.ksh
- aipt_item_rms11.ksh
- aipt_move.ksh
- aipt_orghier_rms11.ksh
- aipt_pad.ksh, aipt_pad_splr.awk, aipt_pad_whse.awk
- aipt_rename.ksh
- aipt_sr0_dayslsld_rms11.ksh

- aipt_str_prd_life_rms11.ksh
- aipt_sub_item_rms11.ksh

RMS 10 Transform Interface Scripts

The execution of these scripts is controlled by running the aip_t_master_rms10.ksh for RMS 10 transforms. These scripts are run before the AIP RPAS batch is run.

For RMS 10, $RPAS_HOME/bin must contain the following scripts:

- aip_t_master_rms10.ksh
- aipt_clnd.awk, aipt_clnd.ksh, days_from_cycle_start_date.pl
- aipt_delete_input_files.ksh
- aipt_future_delivery_rms10.ksh
- aipt_item_rms10.ksh
- aipt_move.ksh
- aipt_orghier_rms10.ksh
- aipt_pad.ksh, aipt_pad_splr.awk, aipt_pad_whse.awk
- aipt_rename.ksh
- aipt_sr0_dayslsld_rms10.ksh
- aipt_str_prd_life_rms10.ksh
- aipt_sub_item_rms10.ksh

Input Schema Files

Location: The schema files matching the RMS RETL extracts are provided by the RMS installation. These files are owned by RMS and are not included with the AIP Package.

Input schema files are the schema files which are required by the Transform Interface scripts to read the RETL extracts from RMS.

These files are used as output schema by the RMS RETL extract scripts.

RMS 14.0, 13.2, 13.1/13.0, and RMS 11 Input Schema Files

For RMS 14.0, 13.2, 13.1/13.0, and RMS 11, the $RMS_ SCHEMA_DIR directory (refer to aip_env_rpas.sh in the *Oracle Retail Advanced Inventory Planning Implementation Guide)* must contain the following input schema files from the RMS installation:

> **Note:** While the file names for these schema files do not differ among RMS versions 14.0, 13.2, 13.1/13.0, and RMS 11, the contents of the files vary among the four versions. Ensure you copy in the correct version of the RMS output schema into $RMS_SCHEMA_DIR.

- rmse_aip_alloc_in_well.schema
- rmse_aip_dmx_bndprdasc.schema
- rmse_aip_future_delivery_alloc.schema
- rmse_aip_future_delivery_order.schema
- rmse_aip_future_delivery_tsf.schema

- rmse_aip_item_loc_traits.schema
- rmse_aip_item_master.schema
- rmse_aip_item_retail.schema
- rmse_aip_item_supp_country.schema
- rmse_aip_merchhier.schema
- rmse_aip_orghier.schema
- rmse_aip_purged_item.schema
- rmse_aip_store.schema
- rmse_aip_substitute_items.schema
- rmse_aip_tsf_in_well.schema
- rmse_aip_wh.schema
- rmse_aip_wh_dat.schema
- rmse_rpas_daily_sales.schema

RMS 10 Input Schema Files

For RMS 10, the $RMS_SCHEMA_DIR directory (refer to aip_env_rpas.sh in the *Oracle Retail Advanced Inventory Planning Implementation Guide)* must contain the following input schema files from the RMS installation:

- rmse_aip_item_master.schema
- rmse_alloc_in_well.schema
- rmse_daily_sales.schema
- rmse_dmx_bndprdasc.schema
- rmse_future_delivery_alloc.schema
- rmse_future_delivery_order.schema
- rmse_future_delivery_tsf.schema
- rmse_item_loc_traits.schema
- rmse_item_retail.schema
- rmse_item_supp_country.schema
- rmse_merchhier.schema
- rmse_orghier.schema
- rmse_purged_item.schema
- rmse_store.schema
- rmse_substitute_items.schema
- rmse_tsf_in_well.schema
- rmse_wh.schema
- rmse_whse1.schema

Output Schema Files

Location: $RMS_AIP_TRANSFORM/aip_schema_dir

Output Schema files are the schema files which are required by the Transform Interface scripts to write the AIP loads/feeds. These files are owned by AIP and should have been unpacked to $RMS_AIP_TRANSFORM/aip_schema_dir during AIP installation.

These output schema files are the same regardless of which version of RMS is interfaced with AIP.

The $AIP_SCHEMA_DIR (refer to aip_env_rpas.sh in the *Oracle Retail Advanced Inventory Planning Implementation Guide)* directory must contain the following output schema files from the AIP installation:

- aipt_dm0_pmsendsrc.schema
- aipt_dm0_pmsstasrc.schema
- aipt_dmx_dscdt_.schema
- aipt_dmx_vadprdasc.schema
- aipt_item.schema
- aipt_loc.schema
- aipt_sr0_dayslsld.schema
- aipt_sr0_it_.schema
- aipt_sr0_oo_.schema
- aipt_str_prd_life.schema
- aipt_wr1_aiw.schema
- aipt_wr1_it_.schema
- aipt_wr1_oo_.schema
- aipt_wr1_tiw.schema

RETL Extracts (Data Files from RMS)

Location: RMS server

The RETL Extracts from RMS are used as AIP loads/feeds. Following are the four categories of RETL Extracts:

Table 3–1 Description of RETL Extracts by Group

Group #	Description of RETL Extracts:
1	The set of extracts that requires transformation before being loaded into AIP RPAS. See these sections: "Needed by AIP RPAS Domain: RMS 14.0, 13.2, and 13.1/13.0 RETL Extracts that are Transformed" on page 3-7 "Needed by AIP RPAS Domain: RMS 11 RETL Extracts that are Transformed" on page 3-9 "Needed by AIP RPAS Domain: RMS 10 RETL Extracts that are Transformed" on page 3-10
2	The set of extracts that does not require any transformation before being loaded into AIP RPAS. See these sections: "Needed by AIP RPAS Domain: RMS 14.0, 13.2, and 13.1/13.0 RETL Extracts that are not Transformed" on page 3-8 "Needed by AIP RPAS Domain: RMS 11 RETL Extracts that are not Transformed" on page 3-9 "Needed by AIP RPAS Domain: RMS 10 RETL Extracts that are not Transformed" on page 3-11
3	The set of extracts which is not used by AIP RPAS but is used by AIP Oracle. See these sections: "Needed by AIP Oracle Database: RMS 14.0, 13.2, and 13.1/13.0 RETL Extracts that are not Transformed" on page 3-8 "Needed by AIP Oracle Database: RMS 11 RETL Extracts that are not Transformed" on page 3-10 "Needed by AIP Oracle Database: RMS 10 RETL Extracts that are not Transformed" on page 3-11
4	The set of extracts which is not used by either AIP RPAS or AIP Oracle. All RMS RETL extracts are deposited to a location on the RMS server. See these sections: "RMS 14.0, 13.2, and 13.1/13.0 RETL Extracts that are not used by AIP" on page 3-8 "RMS 11 RETL Extracts that are not used by AIP" on page 3-10 "RMS 10 RETL Extracts that are not used by AIP" on page 3-11

Note: The four sets of files must be copied from the RMS server to the AIP server, into the $RAW_RMS_DATA_DIR directory.

RMS 14.0, 13.2, and 13.1/13.0 RETL Extracts

The following sections list the RMS 14.0, 13.2, and 13.1/13.0 RETL Extracts for each group described in Table 3–1.

Needed by AIP RPAS Domain: RMS 14.0, 13.2, and 13.1/13.0 RETL Extracts that are Transformed

Following is the list of RETL Extracts that are used by the RMS 14.0, 13.2, and 13.1/13.0-AIP Interface Transform scripts to generate a new set of AIP RPAS loads/feeds.

- date_format_preference.txt
- last_day_of_week.txt
- rmse_aip_alloc_in_well.dat
- rmse_aip_future_delivery_alloc.dat
- rmse_aip_future_delivery_order.dat

- rmse_aip_future_delivery_tsf.dat
- rmse_aip_item_loc_traits.dat
- rmse_aip_item_master.dat
- rmse_aip_item_retail.dat
- rmse_aip_item_supp_country.dat
- rmse_aip_merchhier.dat
- rmse_aip_orghier.dat
- rmse_aip_purged_item.dat
- rmse_aip_store.dat
- rmse_aip_substitute_items.dat
- rmse_aip_tsf_in_well.dat
- rmse_aip_wh.dat
- rmse_aip_wh.txt
- rmse_aip_wh_type.txt
- rmse_rpas_clndmstr.dat
- rmse_rpas_daily_sales.dat

Needed by AIP RPAS Domain: RMS 14.0, 13.2, and 13.1/13.0 RETL Extracts that are not Transformed

The following are the RETL extracts used by RMS 14.0, 13.2, and 13.1/13.0that do not need to be transformed by AIP

- splr.txt
- dmx_dirspl.txt
- dmx_prdspllks.txt
- sr0_curinv_[1..n].txt
- wr1_curinv.txt

> **Note:** As previously shown, the sr0_curinv data feed may be partitioned. For example: sr0_curinv_1.txt, sr0_curinv_2.txt. The data always has at least one partition, namely sr0_curinv_1.txt. AIP RPAS batch steps combine any partitions before loading the data into the AIP domain.

Needed by AIP Oracle Database: RMS 14.0, 13.2, and 13.1/13.0 RETL Extracts that are not Transformed

Following are the RMS 14.0, 13.2, and 13.1/13.0 RETL extracts which are not used by AIP RPAS batch but are used by AIP Oracle batch.

- closed_order.txt
- received_qty.txt

RMS 14.0, 13.2, and 13.1/13.0 RETL Extracts that are not used by AIP

Following are the RMS 14.0, 13.2, and 13.1/13.0 RETL extracts which at this time are not used by AIP.

- rmse_aip_item_supp_country_reject_ord_mult.txt
- rmse_aip_suppliers.dat

RMS 11 RETL Extracts

The following sections list the RMS 11 RETL Extracts for each group described in Table 3–1.

Needed by AIP RPAS Domain: RMS 11 RETL Extracts that are Transformed

Following is the list of RETL Extracts that are used by the RMS 11-AIP Interface Transform scripts to generate a new set of AIP RPAS loads/feeds.

- date_format_preference.txt
- last_day_of_week.txt
- rmse_aip_alloc_in_well.dat
- rmse_aip_future_delivery_alloc.dat
- rmse_aip_future_delivery_order.dat
- rmse_aip_future_delivery_tsf.dat
- rmse_aip_item_loc_traits.dat
- rmse_aip_item_master.dat
- rmse_aip_item_retail.dat
- rmse_aip_item_supp_country.dat
- rmse_aip_merchhier.dat
- rmse_aip_orghier.dat
- rmse_aip_purged_item.dat
- rmse_aip_store.dat
- rmse_aip_substitute_items.dat
- rmse_aip_tsf_in_well.dat
- rmse_aip_wh.dat
- rmse_aip_wh.txt
- rmse_aip_wh_type.txt
- rmse_clndmstr.dat
- rmse_rpas_daily_sales.dat

Needed by AIP RPAS Domain: RMS 11 RETL Extracts that are not Transformed

The following are the RETL extracts used by RMS 11 that do not need to be transformed by AIP.

- splr.txt
- dmx_dirspl.txt
- dmx_prdspllks.txt

- sr0_curinv_[1..n].txt
- wr1_curinv.txt

> **Note:** As previously shown, the sr0_curinv data feed may be partitioned. For example: sr0_curinv_1.txt, sr0_curinv_2.txt. The data always has at least one partition, namely sr0_curinv_1.txt. AIP RPAS batch steps combine any partitions before loading the data into the AIP domain.

Needed by AIP Oracle Database: RMS 11 RETL Extracts that are not Transformed

Following are the RMS 11 RETL extracts which are not used by AIP RPAS batch but are used by AIP Oracle batch.

- closed_order.txt
- received_qty.txt

RMS 11 RETL Extracts that are not used by AIP

Following are the RMS 11 RETL extracts which at this time are not used by AIP.

- rmse_aip_item_supp_country_reject_ord_mult.txt
- rmse_aip_suppliers.dat

RMS 10 RETL Extracts

The following sections list the RMS 10 RETL Extracts for each group described in Table 3–1.

Needed by AIP RPAS Domain: RMS 10 RETL Extracts that are Transformed

Following is the list of RETL Extracts that are used by the RMS 10-AIP Interface Transform scripts to generate a new set of AIP RPAS loads/feeds.

- date_format_preference.txt
- last_day_of_week.txt
- rmse_aip_item_master.dat
- rmse_aip_wh.txt
- rmse_alloc_in_well.dat
- rmse_clndmstr.dat
- rmse_daily_sales.dat
- rmse_future_delivery_alloc.dat
- rmse_future_delivery_order.dat
- rmse_future_delivery_tsf.dat
- rmse_item_loc_traits.dat
- rmse_item_retail.dat
- rmse_item_supp_country.dat
- rmse_merchhier.dat
- rmse_orghier.dat

- rmse_purged_item.dat
- rmse_store.dat
- rmse_substitute_items.dat
- rmse_tsf_in_well.dat
- rmse_wh.dat
- rmse_wh_type.txt

Needed by AIP RPAS Domain: RMS 10 RETL Extracts that are not Transformed

The following are the RMS 10 RETL extracts that do not need to be transformed by AIP.

- splr.txt
- dmx_dirspl.txt
- dmx_prdspllks.txt
- sr0_curinv_[1..n].txt
- wr1_curinv.txt

> **Note:** As previously shown, the sr0_curinv data feed may be partitioned. For example: sr0_curinv_1.txt, sr0_curinv_2.txt. The data always has at least one partition, namely sr0_curinv_1.txt. AIP RPAS batch steps combine any partitions before loading the data into the AIP domain.

Needed by AIP Oracle Database: RMS 10 RETL Extracts that are not Transformed

Following are the RMS 10 RETL extracts which are not used by AIP RPAS batch but are used by AIP Oracle batch.

- closed_order.txt
- received_qty.txt

RMS 10 RETL Extracts that are not used by AIP

Following are the RMS 10 RETL extracts, which for this release are not used by AIP.

- rmse_aip_item_supp_country_reject_ord_mult.txt
- rmse_aip_suppliers.dat

Running the Transform Scripts

This section provides information for running the transform scripts on AIP Oracle and AIP RPAS.

File Transformation and Transfer (AIP RPAS)

The following steps describe how to run the transform scripts for AIP RPAS.

1. Copy Group 1 and Group 2 extracts from the RMS server to the $RAW_RMS_DATA_DIR directory on the AIP server. This process is not part of the Transformation process or AIP batch. It must be scheduled by the client.

2. Copy the RMS RETL Extract schema files for the version of RMS you are running from the RMS installation to the directory specified by the $RMS_SCHEMA_DIR in $RPAS_HOME/bin/aip_env_rpas.sh. Note that previous versions of AIP utilized separate variable names for different versions of RMS. This version of AIP uses a single variable: $RMS_SCHEMA_DIR.
3. Copy the $RMS_AIP_TRANSFORM/aip_schema_dir files from the AIP installation into the $AIP_SCHEMA_DIR. This needs to be done only once per AIP install/patch release.
4. Optionally modify the clnd.prefixes and clnd.day_of_week configuration files, located in the AIP RPAS domain, in $AIPDOMAIN/interface/config/hier, in order to customize the calendar dimension prefixes as well as day of week names.
5. Run the scripts according to the following table:

When extracts are from:	Then run:
RMS 14.0	aip_t_master_rms14.0.ksh
RMS 13.2	RMS 13.2 | aip_t_master_rms13.2.ksh
RMS 13.1/13.0	aip_t_master_rms13.ksh
RMS 11	aip_t_master_rms11.ksh
RMS 10	aip_t_master_rms10.ksh

The location of these scripts is $RPAS_HOME/bin. Either script executes the appropriate RMS-AIP Interface Transform scripts.

Once this script has successfully completed, the following new AIP loads/feeds are created:

clnd.csv.dat	sr0_oo_.txt	wh_type.txt	wr1_oo_.txt
dmx_dscdt_.txt	sr0_dayslsld.txt	whse.txt	wr1_tiw.txt
item.txt	sr0_it_.txt	wr1_aiw.txt	
loc.txt	sr0_prdlfe.txt	wr1_it_.txt	

After the transforms scripts create the new files, there is logic in the scripts to move all files required by AIP RPAS into $INTERFACE_RMS_DIR/input (defined in aip_constants_rpas.sh.) This is the location where the files are processed by AIP RPAS batch. In addition, there is logic in the scripts (see aipt_delete_input_files.ksh) to remove the raw RMS data from the $RAW_RMS_DATA_DIR directory which are not needed by the AIP RPAS domain.

The file clnd.csv.dat is moved to $INTERFACE_RMS_DIR/input and $PURGE_IMPORT_HIER_DIR.

File Transfer for Order Management (AIP Oracle Database)

The client must schedule a batch job to copy Group 3 extracts from the RMS server to the AIP Oracle server location defined by $ONL_INBOUND_DIR in the aip_env_online.sh file. The suffixes for each of the two files (closed_order.txt and received_qty.txt) must be renamed to `.dat`, to create closed_order.dat and received_qty.dat.

The cron_import.sh script does not FTP or copy these two data files to the DM Online server, nor does it uncompress, or untar any package (for example, om.tar.Z)

containing these files. Clients can still use the om.tar.Z file. However, they are required to tar, compress, ftp, uncompress, and untar by using a batch scheduler.

Note: AIP also accepts the OM file rmse_order_purge.txt in this location. However, this file is not critical to AIP—it can be configured as optional—and must be generated from a custom RMS script.

At this point, the RMS-AIP transformation is complete.

RDF to AIP Interfaces

AIP does not provide transformation scripts to format data files from RDF. The data files are expected to arrive in an AIP loadable format. AIP does not provide any script to transfer the RDF data into AIP. You need to have a job scheduled to transfer the files from the RDF server to the AIP RPAS server.

This batch job must copy or FTP the forecast data from the RDF server location to the AIP location, $AIPDOMAIN/interface/forecast.

RDF Extracts

The following table describes the forecast data files that are loaded from the forecasting system into AIP.

Input File Name	Description
iprdfdtdaltv.txt	WRP RDF Alert
sr0_rdfdtdmsk.txt	RDF Detailed Alert Masks
sr0_fcterrlvl1.txt	Store Weekly Demand Forecast Error
sr0_fcterrlvl2.txt	Store Weekly Demand Forecast Error
sr0_rdfdtdcnt.txt	RDF Detail Alert Count
sr0_frclvl1_[1..n].txt	Store Approved RDF Forecast Level 1
sr0_frclvl2_[1..n].txt	Store Approved RDF Forecast Level 2

Once the files are loaded onto the AIP RPAS platform, the AIP RPAS batch script processes the files. The check_load_forecast_data.sh wrapper script is executed to verify that all required files are present before proceeding to load the data into the AIP measures.

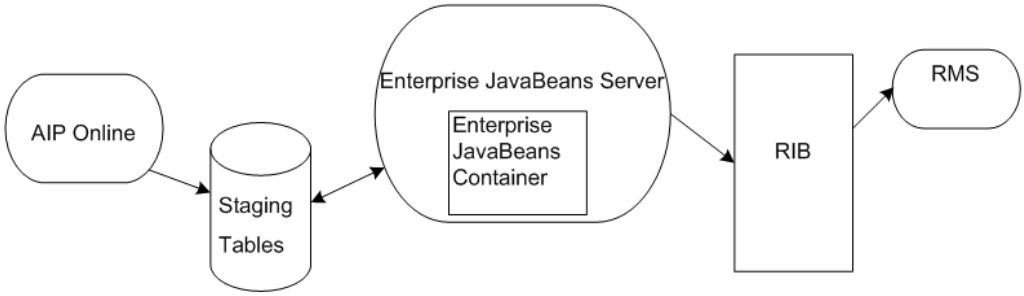
AIP to RMS Interface

The AIP Order Management releases data into the staging tables.

- The polling thread runs EJB at certain time intervals.
- EJB's stateless Session Beans polls data from the staging table and publishes messages to RIB.
- RMS uses RIB to receive messages published by AIP.

Figure 3–1 shows the AIP to RMS process flow.

Figure 3–1 AIP to RMS Process Flow



Message Types

There are five types of messages:

Message Type	Description
XORDERCRE	Create Purchase Order
XORDERDTLCRE	Create Purchase Order Detail
XORDERMOD	Modify Purchase Order
XORDERDTLMOD	Modify Purchase Order Detail
XTSFCRE	Create Transfer

RIB Order Publication

The Oracle Retail Integration Bus (RIB) is a near real-time data synchronization solution used by AIP for publishing orders to RMS. Order publication begins with the order release batch adding the affected order to the appropriate message family queue staging table, and then marking each message with a sequence number. AIP publishes two sets of order messages to the RIB: Purchase Orders and Transfers. RMS subscribes to the RIB messages and inserts the orders into the appropriate RMS Purchase Order and Transfer tables.

AIP Message Flow

A polling operation in the servlet triggers the message creation. The polling is performed by two threads:

- One for the PO_MFQUEUE staging table
- One for the TSF_MFQUEUE staging table

The polling is controlled by the configuration settings in the main.properties file.

The order period count defines the number of time intervals that are to be used. An order period count of 0 indicates that no orders are released. If the order period count is 0, no threads are started.

The time interval defines the amount of time for which the threads sleep. A thread does not go to sleep until less than the maximum number of allowable messages is processed in a given call to the publisher (OrderSenderBean). Publishing less than the maximum allowable messages indicates that all orders on the staging table (at the time it was queried) have been processed. Any orders added to the staging table afterward, are processed the next time the thread wakes up and the publisher is invoked.

For each order period count greater than zero, an order period start and order period end must be added to the properties file. When the thread wakes up and the current time falls between the start and end of any of the intervals (up to X intervals, where X is the order period count), the thread calls the publication procedure. If desired, various time intervals can be created to manage the publication of orders by forcing the threads to poll the staging tables only between certain time periods.

The publisher is an Enterprise Java Session Bean named OrderSenderBean. Its checkAndPublish method queries the staging table and the base order tables to get the message details. The publisher also ensures that the messages are published to the RIB, in the correct order.

Once the message payload is built by the OrderSenderBean, the RIB message publisher takes the payload and wraps it with an envelope used by the RIB infrastructure.

Purchase Order Message

The purchase order publication messages are in the XOrder message family. In AIP, this message family processes the staged orders on the PO_MFQUEUE table.

There are four purchase order message types used by AIP:

- XORDERCRE
- XORDERDTLCRE
- XORDERMOD
- XORDERDTLMOD

All four message types use XOrderDesc.dtd.

XORDERCRE

This message type indicates that a brand new purchase order is being sent to RMS. The orders are sent to RMS in an *A* (Approved) status. This message type is inserted into PO_MFQUEUE in three different circumstances:

The purchase order was released by the batch, or you chose to release the purchase order in the OM Order Maintenance screen.

You created a new purchase order in the OM Order Create screen.

In the OM Order Maintenance screen, you chose to move a purchase order delivery date, or destination, or both, and also generate a new order number.

XORDERDTLCRE

This message type indicates that a new line item is being added to the purchase order after the order was externally communicated. This message type is inserted into PO_MFQUEUE when you have moved the purchase order destination and chosen to retain the existing order number, and the destination does not already exist on the order for that item.

XORDERMOD

This message type indicates that a modification was made to the overall purchase order details (header level information). This message type is inserted into PO_MFQUEUE in the following circumstances:

You have moved the purchase order delivery date and chosen to retain the existing order number.

You have canceled all ordered quantities of all items on the purchase order. The total order quantity for the entire purchase order is zero. The purchase order is sent to RMS with a *C* (Canceled) status.

XORDERDTLMOD

This message type indicates that a modification was made to the purchase order line items after the order was externally communicated. This message type is inserted into PO_MFQUEUE when you perform various actions in the OM Order Maintenance screen:

You modify the order quantity of a purchase order that is not closed.

You chose to move a purchase order line item to a new destination, and also retain the order number. If the move-to destination already exists on the order, a message is written to the staging table to increase the quantity at the move-to location.

> **Note:** Only one message can be inserted for the move-to destination for a particular purchase order. This is either an XORDERDTLCRE if the destination is new, or, XORDERDTLMOD if the SKU is already being delivered to the move-to destination.

The order quantity of the move from destination must be decremented to equal the received quantity. A message is staged for the move from destination.

Transfer Message

The transfer publication messages are in the XTsf message family. In AIP, this message family processes the staged orders on the TSF_MFQUEUE table.

There is one transfer message type used by AIP. It uses XTsfDesc.dtd.

XTSFCRE

This message type indicates that a brand new transfer is being sent to RMS. The transfers are sent to RMS in an *A* (Approved_ status. This message type is inserted into TSF_MFQUEUE when the transfer is released by the batch.

Contingency Overnight Order Processing

The contingency orders in OM give the planners a view of the expected future orders and also serve as backup in the event that new overnight orders cannot be produced from AIP replenishment batch. Since the contingency orders are yesterday's view of today's orders they are very close to the orders that would have likely been produced if today's run of the overnight replenishment batch had completed.

Contingency orders may be available to be released to the RMS in the event that AIP batch cannot produce an order plan, or the data imports into AIP Oracle cannot be completed. The contingency orders, sometimes referred to as yesterday's (forecast) orders, are orders that were generated as forecast orders during the previous batch run. During a normal batch run, yesterday's forecast orders that had not met their release lead time, and had not been released early, are re-planned to ensure the order quantities reflect the latest forecast and inventory information. At the end of the AIP RPAS batch run, the new orders to be release today along with some of the new forecast orders are exported from AIP RPAS to AIP Oracle. During the AIP RPAS replenishment batch calculation of the new order plan, AIP Oracle (OM) still has

yesterday's forecast orders in the database. These orders remain in the Oracle database until the import scripts have been run to load the new orders from AIP RPAS.

It is important to understand that when following the contingency order steps, the contingency orders are a replacement for the orders that would normally be produced in the replenishment batch run for release today. When the batch run is complete, the orders must not be released or executed.

Purchase Order and Transfer Files

The following tables provide a list of the related Purchase Order and Transfer files and what orders are contained in each file.

Table 3–2 Store Orders Purchase Order and Transfer Files

Store Orders	
strsplrord.dat	This file contains store Purchase Orders to be released today. It also contains all Purchase Orders forecasted for the entire planning horizon. The release date in the file indicates whether the order is to be released today, or whether it is a forecast order. Purchase Orders with a release date greater than today are forecast orders. Data in this file is first loaded into interface_store_orders table and then copied into store_order table during the load process.
strwhord.dat	This file contains store Transfers to be released today. The release date in the file equals today. Data in this file is first loaded into interface_store_transfers table during load and then copied into store_order table during the load process.
strsplrord.dat1	This file contains the forecasted store Purchase Orders with an expected release date of tomorrow.
strwhord.dat1	This file contains the forecasted store Transfers with an expected release date of tomorrow.

Table 3–3 Warehouse Orders Purchase Order and Transfer files

Warehouse Orders (Non-contents Orders)	
vendor_to_wh_order.dat	This file contains into-warehouse Purchase Orders to be released today. It also contains all Purchase Orders forecasted for the entire planning horizon. Data in this file is first loaded into warehouse_purchase_order table through i_non_contents_order table during load and then copied into non_contents_order table during merge (that runs before release) process.
wh_to_wh_transfer.dat	This file contains into-warehouse transfers to be released today. It also contains all Transfers forecasted for the entire planning horizon. Data in this file is first loaded into i_non_contents_transfer table during load process. Transfers with release date of today are moved (not copied) into non_contents_order table during release process while transfers with future release dates are copied (not moved) into non_contents_order table during post-release.

Contingency Steps for Overnight Orders

Your operations to perform depend on the point of failure. For your selected failure point, be sure to read and understand all steps before taking action.

Note: The following steps listed will release contingency orders include running cron_release_store_order.sh and cron_release_non_contents_order.sh from cron_release.sh. These are regularly schedule batch scripts that release the orders to be executed. They simply identify all into-store or into-warehouse orders in the Oracle database that have met their lead time.

These scripts must not be run a second time upon successful completion of AIP replenishment batch, otherwise, the system is likely to double order.

Failure Point 1: No Orders Imported from AIP RPAS

If AIP RPAS batch fails before completion of send_scrp_measures_to_online.sh no new RPAS order extracts are available for import into AIP Oracle.

Load Contingency Into-Store Transfers

Perform the following steps to load contingency into-store transfers:

1. In the Oracle schema, navigate to $ONL_INBOUND_DIR.
2. Add an extension, such as today's date, to strwhord.dat in order to prevent the file from being re-loaded into the Oracle table.

   For example, if today's date is 18-January-2015, rename the file to strsplrord.dat.20151801.
3. Rename strwhord.dat1 to strwhord.dat.
4. Navigate to $INTEGRATION_HOME.
5. Execute script cron_import_order.sh with parameters -o transfer -d store. This loads the contingency store transfers found in the strwhord.dat file.

   No action is needed to load any other contingency orders. The contingency store purchase orders and contingency warehouse purchase orders and transfers were loaded during the previous run of cron_import_order.sh.

Release Remaining Orders That Have Met Their Lead Time

Perform the following steps to release remaining orders that have met their lead time:

1. Run cron_release.sh to release today's contingency store purchase orders and transfers.
2. Run cron_release.sh to release today's contingency warehouse purchase orders and transfers.

Note: If failure occurs during send_scrp_measures_to_online.sh, some of the export files may have been generated. If you can verify that some of the files (strsplrord.dat, strwhord.dat, vendor_to_wh_order.dat, wh_to_wh_transfer.dat) have been completely extracted successfully, they should be manually moved to the Oracle database and manually loaded. If strwhord.dat is manually loaded, skip steps 1 through 5 in "Load Contingency Into-Store Transfers" and perform steps 1 and 2 in "Release Remaining Orders That Have Met Their Lead Time".

Failure Point 2: Loading strsplrord.dat

During the run of cron_import_order.sh while loading strsplrord.dat (scripts/import/ord_imp_store_ord_po_in.sh).

If the error is returned by the subscript ord_imp_store_ord_po_in.sh, the contingency purchase orders might have already been deleted in order to prepare the database to import the new purchase order plan.

1. Manually delete any orders from the STORE_ORDER table that was partially loaded as a result of executing ord_imp_store_ord_po_in.sh. These orders can be identified as the orders that are in status *U*, release_date = VDATE, order number IS NULL and source_type = V.
2. Navigate to $ONL_INBOUND_DIR. Make a backup of strsplrord.dat1 by renaming it strsplrord.dat1bak. This is restored at the end of the contingency process.
3. Make a backup of strsplrord.dat by renaming it strsplrord.datbak.
4. Navigate to $BSA_ARCHIVE_DIR. Locate the srp.tar.Z file archived with yesterday's date. Extract the strsplrord.dat1 file and copy it to $ONL_INBOUND_DIR.

   Note: Information about Batch Script Architecture (BSA) previously found in this guide is now in the *Oracle Retail Planning Batch Script Architecture (BSA) Implementation Guide*.

5. Navigate to $ONL_INBOUND_DIR and rename strsplrord.dat1 to strsplrord.dat.
6. Navigate to $INTEGRATION_HOME. Execute: cron_import_order.sh-o purchase -d store.

Release Orders That Have Met Their Lead Time

Perform the following steps to release orders that have met their lead time:

1. Navigate to $INTEGRATION_HOME and execute cron_release.sh to release store purchase order.
2. Restore backup files.

   Navigate to $ONL_INBOUND_DIR. Remove strsplrord.dat. Rename/move strsplrord.datbak to strsplrord.dat. Rename strsplrord.dat1bak to strsplrord.dat1.

   Note: This process only reloaded one day's worth of purchase orders. In the event that the Corrupt Data File process is needed in tomorrow's batch run, no contingency purchase orders are available.

   It is also important to note that the contingency file (.dat1) contains any forecasted purchase orders that were released early. If any purchase orders that are scheduled for release today are executed early (during the previous Online day), a duplicate order is released today.

Failure Point 3: Loading strwhord.dat

During the run of cron_import_order.sh while loading strwhord.dat (scripts/import/ord_imp_store_ord_tsf_in.sh):

1. Manually delete any orders from STORE_ORDER table that were partially loaded as a result of executing ord_imp_store_ord_tsf_in.sh. These orders can be identified as the orders that are in status *U*, release_date = VDATE, release_wave IS NULL, order number IS NULL and source_type = W.
2. Navigate to $ONL_INBOUND_DIR. Make a backup of strwhord.dat1 by renaming it strwhord.dat1bak. This is restored at the end of the contingency process.
3. Make a backup of strwhord.dat by renaming it strwhord.datbak.
4. Navigate to $BSA_ARCHIVE_DIR. Locate the srp.tar.Z file archived with yesterday's date. Extract the strwhord.dat1 file and copy it to $ONL_INBOUND_DIR.
5. Navigate to $ONL_INBOUND_DIR and rename strwhord.dat1 to strwhord.dat.
6. Navigate to $INTEGRATION_HOME. Execute cron_import_order.sh -o transfer -d store. This loads the contingency transfers produced yesterday.

Release Orders That Have Met Their Lead Time

Perform the following steps to release orders that have met their lead time:

1. Navigate to $INTEGRATION_HOME and execute cron_release.sh to release store transfer orders. Restore Backup Files
2. Navigate to $ONL_INBOUND_DIR. Remove strwhord.dat. Rename/move strwhord.datbak to strwhord.dat. Rename strwhord.dat1bak to strwhord.dat1.

Failure Point 4: Loading vendor_to_wh_order.dat into warehouse_purchase_order table

This failure point can happen during the run of cron_import_order.sh (load) while loading vendor_to_wh_order.dat (scripts/import/ord_imp_non_cont_ord_fcst_in.sh) into warehouse_purchase_order table.

If the error was technical in nature and can be fixed, and the interface table i_non_contents_order_forecast still has the data, then rerunning the procedure retl_load.import_wh_purchase_order should be attempted to load the orders.

No contingency file is available in the AIP release. The vendor_to_wh_order.dat file archived in yesterday's srp.tar.Z contains contingency orders. However, it also contains the orders that were released yesterday. The orders which were released yesterday must be deleted from the file before it can be reloaded (including any orders that were released early.)

Failure Point 5: Loading wh_to_wh_transfer.dat into i_non_contents_transfer table

During the run of cron_import_order.sh while loading wh_to_wh_transfer.dat (scripts/import/ ord_imp_non_cont_tsf_in.sh)

No contingency file is available in the AIP release. The wh_to_wh_transfer.dat file archived in yesterday's srp.tar.Z contains contingency orders however it also contains the orders that were released yesterday. The orders which were released yesterday must be deleted from the file before it can be reloaded (including any orders that were released early).

Failure Point 6: Merging Warehouse Purchase Orders into non_contents_transfer table

This failure point can happen during the run of merge_order.sh.

If the failure happens after removing the contingency orders in non_contents_order, then orders should be manually copied from warehouse_purchase_order table into non_contents_order using a similar INSERT query as used in order_merge.copy_order_xxx function.

If the failure happens before removing the contingency orders (of previous merge run) in non_contents_order, then unreleased orders should be manually deleted from non_contents_order.

The script should then be attempted again to merge orders into non_contents_order table.

4

AIP RPAS Batch Processing

The complete AIP batch processing is comprised of the following:

- AIP RPAS batch processing
- AIP Java/Oracle batch processing

Because the AIP solution resides on two platforms and needs to exchange information across both the platforms, the batches are executed on both the platforms.

The aip_batch.sh script is provided with the AIP installation, and it is used to run the entire RPAS daily batch process from a UNIX scheduler. The scripts called by aip_batch.sh may have multi-threaded process calls; however, all the scripts called directly by aip_batch.sh are run inline. The top level control script does not run any scripts in parallel. Therefore, it should be used only as a guide for individual script calls/jobs that function similarly, but cannot be scheduled to run in parallel.

The AIP RPAS Batch Control Script (aip_batch.sh)

The AIP RPAS batch script, aip_batch.sh, accepts a number of arguments that allow more control over what portion of the batch scripts are run. Step names have been defined, which may also be passed into the script as arguments that define exactly which steps and corresponding scripts should be run.

Usage

The following table provides descriptions of the aip_batch.sh arguments.

Argument	Description
-f	First AIP RPAS batch. A predetermined set of steps are run. All other flags can also be used with this flag to further limit the steps that are run. The predetermined skip-steps that should not be run during the first run AIP RPAS batch (as indicated by the -f) is not run, regardless of what arguments are passed.
-s	Indicates that a starting step is defined. The step at which the batch should start must follow this flag. This flag can be used with all the other flags. However, only one batch step can follow the flag. This flag must be used along with the -e flag and its associated argument.
-e	Indicates that an ending step is defined. The step after which the batch should end must follow this flag. This flag can be used with all the other flags. However, only one batch step can follow the flag. This flag must be used along with the -s flag and its associated parameter.

The aip_batch.sh Steps

Table 4–1 is a list of valid steps that can be used with the -s and -e flags. These steps can also be passed as parameters to the aip_batch.sh script, in the form of a list of steps (not flagged with -s and -e). Each step is described in detail in either the sections for "Initial Batch Run" or the "Daily Batch Run". Steps that are run during both the initial and daily batch runs are detailed in the Daily Batch Run column.

Execution Sequence of the AIP RPAS Batch Scripts

Table 4–1 describes the steps in the order that aip_batch.sh execute them.

Table 4–1 Execution Sequence of the AIP RPAS Batch Scripts

Step Name	Description of Action	Initial Batch Run	Daily Batch Run
set_implementation_parameters	Sets all the AIP RPAS implementation parameters.	Yes	No
check_process_external_data	This is a wrapper script for _check_for_required_ files.sh and process_external_data.sh. It first verifies the existence of all the required files as specified by the earlyfiles.config file, and then calls process_external_data.sh.	Yes	Yes
create_empty_default_files	This script creates empty hierarchy files necessary during the first run of the batch program, when the AIP Online data is not available. These empty files are later used for merging with the RMS hierarchy files during the daily batch run.	Yes	No
prep_onl_data	Prepares the daily export files from AIP Online for loading into the AIP RPAS platform.	No	Yes
merge_hierarchies	This script merges the hierarchy data from AIP Online with the hierarchy data from RMS.	Yes	Yes
convert_hierarchies_for_loading	Converts the hierarchies from RMS merged with AIP Online, for loading into the RPAS domains.	Yes	Yes
reconfig_domain_partitions	This script is a wrapper script to the RPAS_ HOME/bin utility reconfigGlobalDomainPartitions, to be used within AIP. It checks for the existence of specific configuration files that trigger the addition or removal of partitions to the prod hierarchy.	Yes	Yes
load_all_hierarchies	Loads all the hierarchies in the AIP domain. This script uses environment variables to determine the domain paths.	Yes	Yes
load_onl_data	Loads the data extracted from the DM Online and OM Online (AIP Oracle) databases.	No	Yes
load_rms_dm_data	Preprocesses and loads all the DM measures that come from RMS.	Yes	Yes
create_empty_archive_files	This script creates new item alerts for all the items.	Yes	No
create_alerts	Calls load_one_newitem_alert_measure.sh for each new item alert measure, and then saves the hierarchy load files for the next time.	Yes	Yes
load_non_rms_external	This script looks for *.ovr files from a non-RMS external system, in the directory specified in the input parameter. The files are moved to the $DOMAIN/input directory, where loadmeasure runs on the files.	Yes	Yes

Table 4–1 (Cont.) Execution Sequence of the AIP RPAS Batch Scripts

Step Name	Description of Action	Initial Batch Run	Daily Batch Run
auto_commit_wkbooks_batch	This script commits all the workbooks (SRP and WRP) on the AIP RPAS platform.	No	Yes
run_partial_dm_batch	Runs the partial Data Management batch to create and assign profiles.	Yes	No
run_dm_batch	This script calls the scripts for the DM critical and noncritical paths of execution.	No	Yes
export_dm_data	Extracts all the DM hierarchy and measure data required to keep DM RPAS and DM Online in sync.	Yes	Yes
check_import_online_schedule	This script imports output from AIP-Online schedules into RPAS-side AIP domain.	No	Yes
dmb_master_post	This script executes dmb_master_post_local.sh on each of the local domains.dmb_master_post_ local.sh in turn sequentially executes the rule groups to be run after importing the schedule from AIP-Online.	No	Yes
check_load_forecast_data	This script is basically a wrapper for load_non_ rms_files.sh, as it pertains to the loading of the forecast data. It first verifies the existence of all the required files as specified by the forecastdata_from_ external.config file. It then calls load_non_rms_ files.sh.	No	Yes
purge_low_variability_advance	Purges and advances the effective date encoded measures along the time dimension, according to the command xml file, purgeLowVariabilityAdvance.xml.	No	Yes
purge_truncate_history	Purge and Truncate History of Selected Replenishment Measures.	No	Yes
copy_sister_data	Copies the sister store and warehouse data by running the command xml files, copySisterStore.xml and copySisterWarehouse.xml.	No	Yes
check_process_inventory_data	This script calls _check_for_required_files.sh and process_inventory_data.sh. It first verifies the existence of all the required files as specified by the latefiles.config file, and then calls process_inventory_data.sh.	No	Yes
load_replenishment_data	Preprocesses and loads all the SRP and WRP measures that come from RMS in the late data feed.	No	Yes
run_replenishment	Runs the replenishment and reconciliation actions, across the supply chain.	No	Yes
export_replenishment_data	Creates the replenishment plan extracts at the local domain level.	No	Yes
send_replenishment_to_online	Concatenates .dat files from an AIP domain's local domain output directories into combined export files that are located in the global domains.	No	Yes
post_dcrp_import	This script imports output from Cross Dock Constraint Receipt Plan(DCRP) into RPAS-side AIP domain.	No	Yes
post_ocs_import	Loads plan adjustments after Scaling is complete.	No	Yes

Table 4–1 (Cont.) Execution Sequence of the AIP RPAS Batch Scripts

Step Name	Description of Action	Initial Batch Run	Daily Batch Run
run_replenishment_post_ processing	Executes the replenishment post-process by sequentially running the defined rule group.	No	Yes
run_dm_alerts	Executes the dmb_master20.sh ${AIPDOMAIN}'script to create alerts, if any.	Yes	Yes
export_dm_alerts	Data management alerts are exported to the AIP Online platform.	Yes	Yes
run_srp_alerts	Executes the generation of SRP alerts by running the defined SRP alert rule group.	No	Yes
run_wrp_item_alerts	Executes the generation of WRP alerts by running the defined WRP alert rule group.	No	Yes
run_wrp_network_alerts	Executes the generation of WRP network alerts by running the defined WRP network alert rule group.	No	Yes
run_replenishment_alerts_post_ processing	Finds generated WRP network alerts and runs post WRP network alerts for global history maintenance.	No	Yes
auto_build_wkbooks_batch	Builds all the SRP and WRP workbooks that have been configured for automatic builds.	No	Yes
run_reports_calculation	Runs all calculations required for OBIEE Reports.	No	Yes
run_dashboard	This script calculates data for AIP Dashboard.	No	Yes

Example Script Calls

Table 4–2 lists example script calls and their actions.

Table 4–2 AIP RPAS Batch Example Script Calls

Command	Action
aip_batch.sh -f	Runs a subset of the `aip_batch.sh` steps. These are the steps that should be run on the very first run of the AIP RPAS batch. See "Initial Batch Run" on page 10-2.
aip_batch.sh -s check_process_external_data -e load_non_rms_ external	Runs the aip_batch.sh steps, starting with check_process_external_data, and up to and including load_non_rms_external.
aip_batch.sh run_dm_alerts	Only runs the aip_batch.sh step listed which is run_dm_alert.
aip_batch.sh run_dm_alerts run_srp_alerts auto_build_ wkbooks_batch	Results in the `aip_batch.sh` running run_dm_ alerts, followed by run_srp_alerts and auto_ build_wkbooks batch.

5

AIP RPAS Daily Batch Scripts

Functionally the AIP batch scripts are broadly classified in Table 5–1. Each of the scripts listed next to its descriptive step name exists to perform that batch step.

Table 5–1 AIP Batch Scripts by Class

AIP Batch Class	Step Description	Batch Script
Set Implementation Parameters	Set Implementation Parameters for DM and Replenishment	set_implementation_parms.sh
External Integration Data Processing	Verify and Process Foundation Data from External System	check_process_external_data.sh
	Create Empty Hierarchy files	create_empty_default_files.sh
Internal Integration Data Processing	Prepare Data from AIP Online platform	prep_from_aiponline.sh
Process Merchandise System and AIP Online Hierarchies	Merge Hierarchies	merge_hierarchies.sh
	Convert Hierarchies for Loading	convert_hierarchies_for_loading.sh
	Reconfigure AIP Domain Partitions	reconfigAIPDomainPartitions.ksh
Load Hierarchy and Non-Inventory Measure Data into AIP RPAS	Load All Hierarchies	load_all_hierarchies.sh
	Load AIP Online Measure Data	load_online_measures.sh
	Load External Non-Inventory Measure Data	load_rms_dm_measures.sh
Generate and Load New Item Alert Measures	Create Empty Archive File	create_empty_archive_files.sh
	Generate Batch and Online Alerts	load_all_newitem_alert_measures.sh
Load External System Measure Data into AIP RPAS	Load Non-RMS External Files	load_non_rms_external.sh
Commit Workbooks before Batch Run	Auto Commit Workbooks	workbook_batch.sh COMMIT

Table 5–1 (Cont.) AIP Batch Scripts by Class

AIP Batch Class	Step Description	Batch Script
Perform and Export Data Management Calculations	Run Initial Load DM Batch	run_partial_dm_batch.sh
	Run DM Batch	dmb_master.sh
	Export DM Data	export_dm_data.sh <true | false>
	Import schedules from AIP-O into AIP-R	check_import_online_schedule.sh
	Run post schedule import DM batch	dmb_master_post.sh
External Integration Forecast Data Processing and Load into AIP RPAS	Check and Load Forecast Data	check_load_forecast_data.sh
Prepare Replenishment Data and Maintain Histor	Purge and Advance Low-Variability Data	purge_low_variability_advance.sh
	Purge Truncate History	for_each_local_domain.sh -p purge_truncate_ history.sh.sh [DOMAIN]
	Copy Sister Stores and Warehouses	for_each_local_domain.sh -p copy_sister_data_local.sh [DOMAIN]
External Integration Inventory Data Processing and Load into AIP RPAS	Verify and Process Inventory Data from External System	check_process_inventory_data.sh
	Load Replenishment Inventory Data	load_replenishment_measures.sh
Calculate and Export Replenishment Plan	Run Replenishment	scrp.sh
	Export Supply Chain Replenishment Data	for_each_local_domain.sh -p export_scrp_inter_meas_ local.sh [DOMAIN]
	Package Supply Chain Replenishment Data	send_scrp_measures_to_online.sh
Import Cross Dock Constraint Receipt Plan (DCRP) output into AIP-R	Import DCRP output into AIP-R domain	post_dcrp_import.sh
Load the Scaled Replenishment Plan	Run Post Supplier and Container Scaling Import	post_ocs_import.sh
Perform Post-Replenishment Calculations	Run Replenishment Post-processing	for_each_local_domain.sh -p scrp_post_local.sh [DOMAIN]
Calculate and Export Data Management Alert	Run Non-critical Data Management Alerts	dmb_master_alerts.sh <true | false>
	Export DM Alerts	export_dm_alerts.sh <true | false>
Compute Replenishment Alerts	Run SRP Item Alerts	scrp_srp_alerts.sh
	Run WRP Item Alerts	wrp_item_alerts.sh
	Run WRP Network Alerts	wrp_network_alerts.sh
	Run Replenishment Alerts Post Processing	replenishment_alerts_post_processing.sh

Table 5–1 (Cont.) AIP Batch Scripts by Class

AIP Batch Class	Step Description	Batch Script
Build Workbooks after Batch Run	Auto Build Workbooks	workbook_batch.sh BUILD
Run Calculations for OBIEE Reports	Calculate Data for OBIEE Reports	run_OBIEE_reports.sh
Calculate Dashboard Measures	Calculate AIP Dashboard data	run_dashboard.sh

Set Implementation Parameters

Step Name

set _implementation_parameters

Script Call

set_implementation_parms.sh

Functional Overview

This script is called to set configurable parameters for DM and replenishment. All parameters are set in measure data and are initialized by running mace expressions using values defined in the shell script aip_env_rpas.sh.

Technical Details

The script set_implementation_parms.sh calls the scripts set_implementation_parms_dm.sh and set_implementation_parms_aip.sh. Each of the scripts is a simple list of rules to be executed as mace commands on the global $AIPDOMAIN. The values of the measures are taken from the configurable shell script aip_env_rpas.sh.

Day on Day Processing

This script performs these steps for day on day processing:

1. Call set_implementation_parms_aip.sh to execute its list of expressions through mace.
2. Call set_implementation_parms_dm.sh to execute its list of expressions through mace.

This Script Calls these Scripts:

- set_implementation_parms_dm.sh
- set_implementation_parms_aip.sh

Appropriate Batch Run (First Time, Daily, or Both)

First Time or ad hoc if a desired value changes.

Prerequisites

Values for all implementation parameters listed in aip_env_rpas.sh must be edited to meet the business requirements of the client.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

External Integration Data Processing

To begin, the data must be exported from RMS or another external system. AIP is integrated with an external system by means of this series of scripts and refers to this series of scripts as Integration. The export of this data is not part of the AIP data export scripts; however, AIP is dependent on the data being available before AIP can proceed with data population. The foundation and inventory data from RMS or another external system are loaded into RPAS as part of the AIP data load process.

Verify and Process Foundation Data from External System

Step Name

check_process_external_data

Script Call

check_process_external_data.sh

Optional Parameters

-h: to display script usage

Functional Overview

RMS is an enterprise solution that provides merchandise hierarchy and organizational setup and maintenance. This data, together with non-RMS external system data, becomes the foundation for the AIP supply chain replenishment. The data is processed as a set of flat files that follow an agreed AIP file format. If the client does not use RMS, but rather some other external system, this data is still required in the same format. AIP validates all of the files to ensure that the required data is present, and massages the files to produce the input required for subsequent steps of the AIP RPAS batch. AIP divides the integration data into two categories: early files and late files. The batch schedule for AIP may aim to maximize the available time by moving as much workload off the critical path as possible. For this reason the static data is extracted from RMS, or similar system, before the dynamic transaction data is available. This allows the hierarchies to be loaded into AIP and the DM batch run before the dynamic data is extracted. This step processes the early, static data.

Technical Details

The static hierarchy and measure data are listed in Table 5–2, Table 5–3, and Table 5–4. For AIP RPAS batch to properly process external system data, all of the flat files must follow a particular format. The flat files should have a *.txt extension.

Table 5–2 lists files that are verified and processed inside the check_process_external_data.sh script, which if the client has RMS and has installed the RMS-AIP Transformation Scripts, are outputs from that transformation process. If the client does not have RMS, then these files are still required. These data are loaded into the RPAS domain as either hierarchy or measure data.

Table 5–2 Early Dynamic and Reference Data

File Name	Explanation
item.txt	Product Hierarchy
loc.txt	Location Hierarchy
splr.txt	Supplier Hierarchy
whse.txt	Warehouse Hierarchy
dmx_dirspl.txt	Direct Suppliers
dmx_dscdt_.txt	Corporate Discontinuation Date
dmx_prdspllks.txt	Commodity-Supplier Links

Table 5–3 lists additional files processed by the check_process_external_data.sh script. However, these are not possible outputs from the RMS-AIP Transformation. These data are also loaded into the RPAS domain as either hierarchy or measure data.

Table 5–3 Load-Ready External Systems Data

File Name	Explanation
had.txt	Advertising Hierarchy
intv.txt	Interval Hierarchy
dm0_ofseffdt_.txt	Off-Sale Effective Date
dm0_onseffdt_.txt	On-Sale Effective Date
dmx_shpto_.txt	Receiving Supplier / Ship To
ipavgrtslsi.txt	Total Store Average Rate Of Sales
iphldbckqtyi.txt	Hold Back Quantity
ipiavgrtslsi.txt	WH Independent ARS
ipldssi.txt	Loaded Safety Stock
ipodcmti.txt	Order Commit
iprpltcdi.txt	Replenishment Type Code
iprplstcdi.txt	Replenishment Subtype Code
ipslsi.txt	Historical Weekly Sales
sr0_ad_.txt	Store Ads
sr0_ad_go_.txt	Store Ads Grand Opening
sr0_ad_irt.txt	Store Ads Inserts
sr0_ad_oth.txt	Store Ads Others
sr0_ad_rop.txt	Store Ads Run on Press
sr0_adjsls.txt	Store Adjusted Sales
sr0_avgrosld_.txt	Store Average Weekly Rate of Sale Loaded
sr0_co_.txt	Store Customer Orders
sr0_hstls_.txt	Store Historical Lost Sales
sr0_knowndemand.txt	Store Known Demand
sr0_ovrstkflg.txt	Run Overstock Alert Flag

Table 5–3 (Cont.) Load-Ready External Systems Data

File Name	Explanation
sr0_rplcde.txt	Store Repl Type Code
sr0_rplsubcde.txt	Store Repl Subtype Code
sr0_ss_ld_.txt	Store Loaded Safety Stock
sr0_tdgday.txt	Store Trading Days
srx_prdrpr.txt	SKU Retail Price
ipfctwkprfd.txt	Week to Day Forecast Percentage Default (Un-Normalized)
ipfctwkprfe.txt	Store Week to Day Forecast Percentage Override (Un-Normalized)
ipwhhldcpci.txt	Stocking Point Holding Capacity
srx_poidst.txt	Poisson Distribution Lookup Table
sr0_wkbsf_ld.txt	Loaded Weekly Base Sales Forecast
ipttlhlstki.txt	Total Held Stock
ipadstai.txt	Store Ad Start Date
ipadendi.txt	Store Ad End Date
sr0_dyscsls.txt	Daily Short Code Sales
sr0_invadj.txt	Inventory Adjustments
sr0_wstadj.txt	Waste Adjustments
ipibcpcco.txt	Inbound Capacity Cases for Reporting
ipobcpcco.txt	Outbound Capacity Cases for Reporting
ipcurinvcominiti.txt	Loaded Store Current Inventory by Product Expiration Date
ipoocomi.txt	Loaded Store On Orders Composition by Product Expiration Date
ipitcomi.txt	Loaded Store In Transit Composition by Product Expiration Date
ipextdmdo.txt	External Demand
ipotbbdgclsi.txt	OTB Budget by Class/Week
ipotbbdgscli.txt	OTB Budget by Subclass/Week
ipotbavlclsi.txt	OTB Available by Class/Week
ipotbavlscli.txt	OTB Available by Subclass/Week
ipuntcsti.txt	Unit Financial Value
idrwal1i.txt	Intra-day Release Wave Assignment by Store/Seq/SKU/day
idrwal2i.txt	Intra-day Release Wave Assignment by Store/Seq/SKU/Week-pattern
idrwal3i.txt	Intra-day Release Wave Assignment by Store/Seq/Department/Day
idrwal4i.txt	Intra-day Release Wave Assignment by Store/Seq/Department/Week-pattern
ipddpl1i.txt	Delivery-day demand percent by Location/SKU/Day/Delivery
ipddpl2i.txt	Delivery-day demand percent by Location/SKU/Week-pattern/Delivery

Table 5–3 (Cont.) Load-Ready External Systems Data

File Name	Explanation
ipddpl3i.txt	Delivery-day demand percent by Location/Department/Day/Delivery
ipddpl4i.txt	Delivery-day demand percent by Location/Department/Week-pattern/Delivery
ipddpl5i.txt	Delivery-day demand percent by Location/Day/Delivery
ipddpl6i.txt	Delivery-day demand percent by Location/Week-pattern/Delivery
iphrslsprofi.txt	Hourly Sales Profile
ipddpcurinvl1i.txt	DDP for Current Inventory Exception
ipddpcurinvl2i.txt	DDP for Current Inventory Default
idrplnwavl1i.txt	Intra-day Release Replan Indicator by Store/Seq/SKU/Day
idrplnwavl2i.txt	Intra-day Release Replan Indicator by Store/Seq/SKU/Week-pattern
idrplnwavl3i.txt	Intra-day Release Replan Indicator by Store/Seq/Department/Day
idrplnwavl4i.txt	Intra-day Release Replan Indicator by Store/Seq/Department/Week-pattern
iponshelfl1i.txt	On-shelf time by Store/SKU/Day
iponshelfl2i.txt	On-shelf Time by Store/Department/Day
iponshelfl3i.txt	On-shelf Time by Store/Department/Week-pattern
idsnpshttiml1i.txt	Intra-day Release Wave Inventory Snapshot Time Exception
idsnpshttiml2i.txt	Intra-day Release Wave Inventory Snapshot Time Default

Table 5–4 lists non-load-ready data that is received from an external (non-RMS) system, which is processed by the check_process_external_data.sh script. This data is prepared for intragration transfer to AIP Online, but some of the data is split to create load-ready measure data that is loaded into the RPAS domain.

Table 5–4 Non-Load-Ready External Systems Data

File Name	Explanation
default_wh.txt	Default Warehouse. This data file is split into two load-ready data files: ■ dmx_defwh_.txt / Default Warehouse ■ dmx_defwh_csc.txt. / Default Warehouse Customer Service Center
direct_store_format_pack_size.txt	Direct Store Format Pack Size
direct_store_pack_size.txt	Direct Store Pack Size
store_format_pack_size.txt	Store Format Pack Size
store_pack_size.txt	Store Pack Size
item_attribute.txt	Item Attribute
item_attribute_type.txt	Item Attribute Type

Table 5–4 (Cont.) Non-Load-Ready External Systems Data

File Name	Explanation
sister_store.txt	Sister Store Information. This data file is split into two load-ready data files: ■ dmx_sst.txt / Sister Store ■ dmx_stropndt_.txt. / Store Open Date
sister_wh.txt	Sister Warehouse. This data file is split into two load-ready data files: ■ dmx_wh_.txt / Sister Warehouse ■ dmx_wh_opndt_.txt / Warehouse Open Date
wh_type.txt	Warehouse Type

Day on Day Processing

This script performs these steps for day on day processing:

1. Verify that all required files have been downloaded. Failure to download any of the required files results in an error and termination of the batch.
2. Split the sister_store.txt, sister_wh.txt and default_wh.txt files into load-ready data files as described in the previous tables.
3. Prefix the following measure data with the stocking point prefixes, using construct_ntier_measuredata.ksh:
   - dmx_prdspllks.txt
   - dmx_dirspl.txt
   - wh_type.txt
4. Using interutil binary, $AIPDOMAIN/interface/config/rms/hier/item.config, and the $AIPDOMAIN/interface/config/rms/hier/prod_external.format file, the following hierarchy and measure files are created from item.txt.

File Name	Label
prod.dat	RMS Product Hierarchy
dmx_rmsskumap.ovr	RMS SKU Map
dmx_pszmap.ovr	Pack size Map

   In addition, the interutil program converts from RMS SKU to AIP SKU all RMS-sourced inventory measure data that has been configured to arrive early, and therefore is present to be processed by this script (as opposed to arriving late, and therefore is processed by check_process_inventory_data.sh).

   Note: The dmx_rmsskumap measure data is not loaded into RPAS. The later mapping of RMS SKU to AIP SKU is done without use of this loaded data. However, this data is still required as it is delivered to the AIP Oracle part of the application.

5. Rename prod.dat to prod.txt.
6. Call construct_ntier_hierarchies.sh to add stocking point prefixes to whse.txt, loc.txt, splr.txt, and generate the ssp and dsp hierarchy.dat files.

7. Copy prod.txt and whse.txt from $AIPDOMAIN/interface/rms/input to prod.def and whse.def in $AIPDOMAIN/interface/import for merging in a later step with AIP-Oracle DB product and warehouse contributions. Also copy splr.txt and loc.txt in $AIPDOMAIN/interface/rms/input to hspl.dat and loc.dat in $AIPDOMAIN/interface/import/hier.
8. Move non-RMS, external, measure data listed in $AIPDOMAIN/interface/config/external/measdata_from_external.config from the $AIPDOMAIN/interface/rms/input directory to the $AIPDOMAIN/interface/external directory where they are loaded into the AIP domain through a later script, load_non_rms_files.sh.
9. Move non-RMS, external, non-measure data listed in $AIPDOMAIN/interface/config/external/aiponlinedata_from_external.config to $AIPDOMAIN/interface/export so it can be exported to AIP online through a later script. During this step, prior to moving wh_type.txt, make a copy of wh_type.txt as $AIPDOMAIN/interface/external/dmx_wh_typ.ovr so it can be loaded into the $AIPDOMAIN.
10. Create a copy of hspl.txt named prof.def. In a later step, when Automatic Warehouse Profile Creation is enabled, this file is formatted to match the warehouse profiles created for Suppliers and merged with the existing DM Online Warehouse Profiles.
11. Remove the marker $AIPDOMAIN/interface/export_dm_data, set by send_dm_measures_to_online.sh. This marker indicates a successful export in the export_dm_data.sh call from the previous run of AIP RPAS batch. When marker is not present, the export process exports deltas, and sets the marker. If the marker is present, then export_dm_data.sh preserves a backup of the previous export before generating a new one, in order that a restart/recovery of this step can be implemented.

This Script Calls these Scripts:

- _check_for_required_files (a function defined in bsa_check_for_required_files.sh)
- process_external_data.sh

Appropriate Batch Run (First Time, Daily, or Both)

First Time and Daily.

Prerequisites

This step must not be initiated until the following is successfully completed:

- Before this script runs, all data files from Table 1, Table 2, and Table 3 should be copied by the client's batch scheduler into the $AIPDOMAIN/interface/rms/input directory. Some of the data is required, and some is optional. Reference the $AIPDOMAIN/interface/config/external/earlyfiles.config for requirements.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Move prod.txt, whse.txt, splr.txt, and loc.txt to the $AIPDOMAIN/interface/import directory. The purpose of this is to have

hierarchies ready in AIP Online format, and they should be merged with the actual AIP Online hierarchies at a later step.

4. Re-copy or re-FTP all required merchandising and external system*.txt files into the $AIPDOMAIN/interface/rms/input directory.
5. Restart the batch.

Note: This step can be run in parallel with Export DM and OM Online Data and Retrieve data from AIP Online.

Create Empty Hierarchy Files

Step Name

create_empty_default_files

Script Call

create_empty_default_files.sh

Functional Overview

When AIP Batch is run for the first time, there is no AIP Online data that can be imported into AIP RPAS. Therefore there are no product and warehouse hierarchy files provided by AIP Online to be merged with the product and warehouse hierarchy files provided by an inventory management system. This step creates empty files that act as placeholders for the merge operation.

Technical Details

The script create_default_positions.sh is used to create 0-byte files for prod.dat and whse.dat.

Day on Day Processing

This script performs these steps for day on day processing:

1. Remove $AIPDOMAIN/interface/import/*.dat. This clears out any unwanted, residual data that exist prior to the first-time run.
2. Use UNIX touch command to create $AIPDOMAIN/interface/import/prod.dat and $AIPDOMAIN/interface/import/whse.dat.

Appropriate Batch Run (First Time, Daily, or Both)

First Time only.

Prerequisites

None.

Restart/Recovery

If create_empty_default_files.sh fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Internal Integration Data Processing

AIP RPAS is integrated with AIP Online by means of exporting data from AIP Online into AIP RPAS, and at the end of the AIP RPAS batch process, by exporting data from AIP RPAS into AIP Online. These two exchanges inside AIP are referred to as Intragration. The first export occurs early in AIP RPAS batch to retrieve the AIP Online data into the AIP RPAS domain.

Prepare Data from AIP Online Platform

Step Name

prep_onl_data

Script Call

prep_from_aiponline.sh

Functional Overview

You setup and maintain the supply chain and many replenishment parameters in DM Online. All of this data is used by AIP RPAS batch. For AIP RPAS batch to use this data, it must be retrieved from the Oracle database and loaded into the RPAS domain.

In addition to supply chain data, DM Oracle batch maintains the virtual date used by AIP RPAS batch. This value is loaded into RPAS along with the rest of the DM Online data.

Technical Details

AIP RPAS processes data from AIP Online. AIP Online exports hierarchy information and DM measures which are placed by the user into the AIP RPAS domain, in the $AIPDOMAIN/interface/import directory.

The prep_from_aiponline.sh script is called to process the data files created by the AIP Online export. The data files should be transferred from AIP Online to the export directory by a job scheduling application.

Day on Day Processing

This script performs these steps for day on day processing:

1. prep_files.sh is called two times. The first call processes the AIP Online hierarchy export. The second call processes the AIP Online measure export. The prep_files.sh script is invoked with two keys that are listed inside $AIPDOMAIN/interface/config/bsa_prep_files.config:
    - **DMo_hier_export** is keyed to process the file hierarchy.tar.Z and unpack the data contained within to the $AIPDOMAIN/interface/import/hier directory.
    - **DMo_meas_export** is keyed to process the file aip.tar.Z and unpack the data contained within to the $AIPDOMAIN/interface/import/meas directory.
    - The data unpacked overwrites any existing files in these two directories.
    - Before each archive is unpacked, a backup of the file is created in the $BSA_ARCHIVE_DIR directory. The backup filenames are aip.tar.Z.<timestamp> and hierarchy.tar.Z.<timestamp>.
    - After each archive is unpacked, the archive file is removed from $AIPDOMAIN/interface/import.

2. Verify existence of the unpacked measure data files using the compressed.config, uncompressed.config and lowvariability.config file lists in $AIPDOMAIN/interface/config/meas. If any are missing, halt batch with error.
3. Two data files are moved from $AIPDOMAIN/interface/import/hier to $AIPDOMAIN/interface/import: prod.dat and whse.dat. This is to prepare for the hierarchy merging.
4. Boolean TRUE values are appended to the data that is currently exported from AIP Online. The files are overwritten in the $AIPDOMAIN/interface/import/meas directory. Files currently exported in this way that have Boolean TRUE values appended are currently the following files:
   - dm1_prflks

This script calls the following script:

prep_files.sh

Appropriate Batch Run (First Time, Daily, or Both)

Daily only.

Prerequisites

This step must not be initiated until successful completion of the following:

- The cron_export.sh script runs in the AIP Oracle schema.

Restart/Recovery

If this step fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Ensure that cron_export.sh ran correctly on the AIP Oracle schema. If it did not, perform the restart/recovery steps for cron_export.sh.
4. Ensure that *.Z files were correctly FTPed to RPAS. If not, archive or remove any RPAS zip files under $AIPDOMAIN/interface/import and rerun the client-owned batch process to FTP the data from AIP Oracle side to AIP RPAS side.

   Note: Both the hierarchy.tar.Z and aip.tar.Z files need to exist in the [domain]/interface/rms directory.

5. If cron_export.sh ran correctly, but the prep_from_aiponline.sh fails because of a measure data file or hierachy data file, then determine which data file caused the failure.
   - If the measure data file failed, then run: `prep_from_aiponline.sh -meas`
   - If the hierachy data file failed, then run: `prep_from_aiponline.sh -hier`
6. Restart the batch.

   Note: This step can be run in parallel with check_process_external_data.

Process Merchandising System and AIP Online Hierarchies

Before AIP RPAS can manipulate the measure data, first the new hierarchy data must be loaded. New hierarchy positions can potentially arrive from both the merchandising system (such as RMS) as well as from AIP Online.

Merge Hierarchies

Step Name

merge_hierarchies

Script Call

merge_hierarchies.sh

Functional Overview

After the initial run, AIP RPAS batch loads data from RMS and AIP Online. Various AIP specific attributes of the RMS foundation data are maintained in DM Online. These attributes must be merged with the data that was loaded from RMS before being loaded into the RPAS domains.

Note: AIP Online does not allow the modification of any attributes mastered in RMS, such as warehouse name, SKU name, or department.

Technical Details

This step merges the AIP Online hierarchies with the external system hierarchies (RMS). At a later time, the merged hierarchies are loaded into the AIP RPAS. The merge_hierarchies.sh script is used to perform the merge.

Day on Day Processing

This script performs these steps for day on day processing:

The prod.dat and whse.dat files have default positions, which are later assigned through AIP Online. Therefore, these files need to be merged with the RMS hierarchies. AIP Online does not affect the location and supplier hierarchy files. Therefore, the RMS contributions of loc.txt and hspl.txt are moved to be loaded directly into AIP RPAS domains.

1. The hierarchies are merged using run_interutil.sh. The run_interutil.sh uses the interutil binary and hierarchy configuration files to merge the online hierarchy with the RMS hierarchy and to generate the final hierarchy file.
   - prod.dat + prod.def using prod.config generates prod.dat
   - whse.dat + whse.def using whse.config generates whse.dat

   All of these outputs are in AIP Online format and need to be converted to AIP RPAS format before getting loaded into RPAS.
2. Call merge_prof.sh to merge prof.def from External Source and prof.dat from AIP Online.
3. Remove pre-merge files from $AIPDOMAIN/interface/import: prod.def, prod.dat, whse.def, whse.dat.

This Script Calls these Scripts:

- run_interutil.sh
- merge_prof.sh

Prerequisites

All steps documented previously in this guide.

Restart/Recovery

If this step fails, perform the following:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Ensure that the RMS data files (prod.def and whse.def) exist in the $AIPDOMAIN/interface/import directory with the correct format.

   If they do not exist, re-copy $AIPDOMAIN/interface/rms/prod.txt to $AIPDOMAIN/interface/import/prod.def, and $AIPDOMAIN/interface/rms/whse.txt to $AIPDOMAIN/interface/import/whse.def.
4. Ensure that the AIP Online files (prod.dat and whse.dat) exist in the $AIPDOMAIN/interface/import directory with the correct format.

   If they do not exist, extract prod.dat and whse.dat from $BSA_ARCHIVE_DIR/hierarchy.tar.Z.<timestamp> and place them in $AIPDOMAIN/interface/import.
5. Restart the batch.

Convert Hierarchies for Loading

Step Name

convert_hierarchies_for_loading

Script Call

convert_hierarchies_for_loading.sh

Functional Overview

This script preprocesses a subset of the hierarchy load files, prepending position names to the existing label field text.

Technical Details

The following hierarchy load files are preprocessed to prepend the hierarchy position names to the position labels:

- prod
- loc
- whse
- hspl
- ssp
- dsp

The prepending process is handled in a set of awk scripts.

The converted hierarchy data files and all other hierarchy files are copied to the $AIPDOMAIN/input directory.

Day on Day Processing

This script performs these steps for day on day processing:

1. Remove any existing hierarchy load files, $AIPDOMAIN/input/*.dat.
2. Copy from $AIPDOMAIN/interface/import/hier to $AIPDOMAIN/input all files (matching the pattern *.dat). This copies the following files:
   - Location and Supplier hierarchy data files received from merchandising system.
   - Network Group, Order Lead Time Cycle, Order Group, and Profile Order Cycle hierarchy data received from AIP-Oracle.
   - Product, Warehouse and Profile hierarchy data files merged from merchandising system and AIP-Oracle.
3. Move the Advertising and Interval hierarchy data files (if they exist), and the Source and Destination Stocking Point hierarchy data files from $AIPDOMAIN/interface/rms to $AIPDOMAIN/input.
4. Copy the calendar (clnd.dat or clnd.csv.dat, whichever is in $AIPDOMAIN/interface/rms) into $AIPDOMAIN/input.
5. Merge the position name into the label for the product, location, warehouse, supplier, source and destination stocking point hierarchies, so that when viewed through the RPAS Client, the position name is displayed in the label.
6. Copy the product, location, supplier and warehouse hierarchy files from $AIPDOMAIN/input to $AIPDOMAIN/interface for processing by the new hierarchy alert logic later in the AIP RPAS batch.

This Script Calls these Scripts:

- convert_hierarchies_to_csv_for_load.sh
- mergeProdPositions.awk
- mergeStrPositions.awk
- mergeWhPositions.awk
- mergeSplrPositions.awk
- mergeDspSspPositions.awk

Appropriate Batch Run (First Time, Daily, or Both)

- First Time
- Daily

Prerequisites

This step must not be initiated until successful completion of the following:

- merge_hierarchies.sh

Restart/Recovery

If this step fails, perform the following:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Reconfigure AIP Domain Partitions

Step Name

reconfig_domain_partitions

Script Call

reconfigAIPDomainPartitions.ksh

Functional Overview

This step automates the addition of new partitioning dimension positions to the AIP RPAS domain. When adding these subclasses, it ensures that no local domain is overloaded and spreads the incoming subclasses as evenly as possible. It also ensures that related items, in term of merchandise hierarchy, are clustered together in local domains.

Technical Details

This script is a wrapper script to the script generateReconfigPartDimXml.ksh and the RPAS_HOME/bin utility reconfigGlobalDomainPartitions to be used within AIP. The script generateReconfigPartDimXml.ksh creates reconfigpartdim.xml if any new subclass is coming from RMS in the prod.dat file.

The xml file reconfigpartdim.xml may be created during this process. This file is created if a new subclass is found in the prod.dat which is not part of the existing domain structure. The RPAS utility reconfigGlobalDomainPartitions use the auto-created reconfigpartdim.xml to add the new subclass positions to the domain.

In a first day run, the call to generateReconfigPartDimXml.ksh is bypassed. It is assumed that on the first day, the subclasses loaded into the domain during domain creation are in synchronization with the subclasses in the product hierarchy loaded from merchandising system and therefore there are no new subclasses.

In an every-day run, if logic in this step determines that all subclasses are new, then generateReconfigPartDimXml.ksh will not generate reconfigpartdim.xml file and no reconfiguration will be run. All new subclasses will be placed in the default domain (as specified by $DEFAULT_DOMAIN variable in $RPAS_HOME/bin/aip_env_rpas.sh) during the load_all_hierarchies step of aip_batch.sh.

To remove positions along the partitioning dimension, use the AIP RPAS Purging functionality described in the AIP Interval Batch Scripts chapter of this guide.

Day on Day Processing

This script performs these steps for day on day processing:

1. Call generateReconfigPartDimXml.ksh to search $AIPDOMAIN/input/prod.dat, looking for new partitioning dimension positions. If any is found, generate $AIPDOMAIN/config/reconfigpartdim.xml.
2. If both of the following files exist:
   - $AIPDOMAIN/config/reconfigpartdim.xml
   - $AIPDOMAIN/input/prod.dat

Then run dimension addition by:

- Calling reconfigGlobalDomainPartitions to add the positions indicated in reconfigpartdim.xml.
- Move reconfigpartdim.xml to the $AIPDOMAIN/input/processed directory and append a timestamp to the file.

Appropriate Batch Run (First Time, Daily, or Both)

- First Time
- Daily

Prerequisites

This step must not be initiated until successful completion of the following:

- convert_hierarchies_for_loading.sh

Restart/Recovery

If this step fails, perform the following:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Load Hierarchy and Non-Inventory Measure Data into AIP RPAS

The following sections describe the process of Load Hierarchy and Non-Inventory Measure Data into AIP RPAS.

Load All Hierarchies

Step Name

load_all_hierarchies

Script Call

load_all_hierarchies.sh

Functional Overview

This script loads all hierarchy load files into the global and local domains.

Technical Details

All hierarchy load files are loaded into AIPDOMAIN and its local domains by standard RPAS functionality.

Day on Day Processing

Call loadHier to load all hierarchy load files found in the $AIPDOMAIN/input directory.

Appropriate Batch Run (First Time, Daily, or Both)

- First Time
- Daily

Prerequisites

This step must not be initiated until successful completion of the following:

- reconfigAIPDomainPartitions.ksh

Restart/Recovery

If this step fails, perform the following:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. If the script failed to load specific positions, identify from where these positions are coming.
4. If new positions are coming from RMS, identify them in item.txt. If the data coming from RMS is incomplete, remove that line and raise a defect to the RMS team.
5. If new positions are coming from DM Online with erroneous data or bad formatting, remove data from the load file and raise a defect for DM Online.
6. If there are no new positions coming from either system, recopy from $AIPDOMAIN/input/processed to $AIPDOMAIN/input the previous successful hierarchy loads. Remove the timestamp suffix so that the suffix is .dat.
7. Restart the batch.

Note: If the files are in the correct format, you should be able to run the loadHier RPAS utility on individual files without rerunning the entire step.

- loadHier
- -d <Global Domain> -loadAll -checkParents -maxProcesses ${BSA_MAX_PARALLEL}-forceInputRollups

---

Load AIP Online Measure Data

Step Name

load_onl_data

Script Call

load_online_measures.sh

Functional Overview

Data files from the AIP Online part of AIP are loaded into the AIP RPAS domain.

Technical Details

The data files from AIP Online are listed in two configuration files: $AIPDOMAIN/interface/config/meas contains compressed.config and uncompressed.config. The compressed.config file lists the measure data files that are compressed by time, in that consecutive (by day) equivalent values are not contained in the data for the same intersection (example, SKU/store). The uncompressed.config lists the measure data files that are not compressed by time.

Each measure listed in these two configuration files is contained in the aip.tar.Z file which is transferred from the AIP Online server to the AIP RPAS server, and unpacked in the $AIPDOMAIN/interface/import/meas directory, in the previously described process.

The compressed measure data follows this technical process: the data is loaded into a compressed staging measure, then uncompressed into the actual measure array. The measure data is uncompressed to a different target end date depending on the planning horizon and other functional considerations. The uncompressed measure data is loaded into the measure array without need for staging measure load and uncompressing. In both cases, the measure data is always loaded as an overlay (a load into currently populated data locations overwrites, and all other data remains intact); however, the measure array may be partially or a fully cleared prior to this overlay load, depending on the functional requirements. Neither the compression/uncompression flag, nor the end date for uncompressing, nor the clearing prior to load are configurable by the client.

Day on Day Processing

This script performs these steps for day on day processing:

1. All measure data files listed in the uncompressed.config, compressed.config, and lowvariability.config are moved from $AIPOMAIN/interface/import/meas/*.dat to $AIPOMAIN/input/*.ovr.
2. The measure data deletion flags, which indicate values have been deleted in AIP Online, are converted into actual NA values.
3. The planning horizon measures are loaded.
4. All measure data files listed in the uncompressed.config and lowvariability.config are cleared (if scheduled for clear operation) and loaded into the measure arrays in $AIPDOMAIN.
5. All measure data files listed in the compressed.config are cleared (if scheduled for clear operation) and uncompressed into the measure arrays in $AIPDOMAIN.
6. All measures with a day dimension in their base intersection, but which are not scheduled for future-clear operation, are extended by one day to accommodate the rollover of the actual day.

This script or sub-scripts call the following programs:

- clearArray (AIP binary)
- compressValues (AIP binary)
- loadmeasure (RPAS binary)
- load_measures.sh
- run_interutil.sh (script wrapper for interutil AIP binary)
- xmace (AIP script wrapper for RPAS mace binary)

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful completion of the following:

- All prior steps in this AIP batch process must be completed with the exception of check_process_inventory_data. The data load is dependent on the successful loading of the new hierarchy information from AIP Online merged with External Hierarchy data.

Restart/Recovery

If this step failed, check the following:

1. Examine the log generated to understand why it failed.
2. If the failure was for a specific measures, check which positions within that measures is failing.
3. Ensure that those positions were successfully loaded during the Load All Hierarchies step.
4. If all positions are correctly loaded, check to ensure that the data formats in the *.dat files from Online are correct.
5. If positions were incorrectly loaded during the Load All Hierarchies step, fix it, and re-run loadHier with the correct values for the domain where the failure occurred.
6. Extract the most recent $BSA_ARCHIVE_DIR/aip.tar.Z.<timestamp> into $AIPDOMAIN/interface/import/meas directory.
7. Special processing is required to make dm1_prflks loadable:
   a. Rename $AIPDOMAIN/interface/import/meas/dm1_prflks.dat to tmp_dm1_prflks.dat.
   b. Run the following command:

      sed -e 's/$/1/g' $AIPDOMAIN/interface/import/meas/tmp_dm1_prflks.dat > $AIPDOMAIN/interface/import/meas/dm1_prflks.dat

   c. Remove $AIPDOMAIN/interface/import/meas/tmp_dm1_prflks.dat
8. Restart the batch.

Load External Non-Inventory Measure Data

Step Name

load_rms_dm_data

Script Call

load_rms_dm_measures.sh

Functional Overview

This step preprocesses and loads into the RPAS domain all Data Management measure data that comes from RMS or another external inventory system.

Technical Details

This script loads into $AIPDOMAIN all Data Management measure data in $AIPDOMAIN/interface/rms.

Day on Day Processing

This script performs this step for day on day processing:

1. Call loadmeasure on the measure data files corresponding to the Data Management measure data received from the inventory system (example, RMS). The measures loaded are listed in the following configuration files:
   - $AIPDOMAIN/interface/config/rms/meas/rms_sku_map.config
   - $AIPDOMAIN/interface/config/rms/meas/dm_rms_measures.config

   Note: Some measures are loaded as overlays (.ovr) and some are loaded as full replacement (.rpl). This is not intended to be configurable due to functional requirements.

This script calls the following programs:

- loadmeasure
- exportData

Appropriate Batch Run (First Time, Daily, or Both)

- First Time
- Daily

Prerequisites

This step is dependent on successful completion of the following:

- All steps through the load_all_hierarchies.sh script

Restart/Recovery

If this step fails, you need to check the following:

1. Ensure all DM RMS files listed in the two configuration files are present in $AIPDOMAIN/interface/rms.
   - For those that loaded successfully according to the log files, there is no need to copy them back to $AIPDOMAIN/interface/rms.
   - For those which did not load successfully according to log files, copy them from $AIPDOMAIN/input/processed to $AIPDOMAIN/interface/rms, at the same removing the timestamp and changing suffix to the *source suffix* specified in the two configuration files. The format of the configuration files is: measurename.destination_suffix.source_suffix, e.g. dmx_dscdt_.rpl.ovr.
2. Ensure all files are formatted correctly.
3. Ensure that the Load All Hierarchies step ran correctly, and all of the hierarchies' values received from RMS are loaded correctly.
4. If any of the previous steps did not run correctly, re-run the corresponding step.
5. Restart the batch.

Notes

This step can be run in parallel with Load Measures with AIP Online Data.

Generate and Load New Item Alert Measures

The following sections describe how to Generate and load new item alert measures.

Create Empty Archive Files

Step Name
create_empty_archive_files

Script Call
create_empty_archive_files.sh

Functional Overview
When AIP Batch is run for the first time, there are no hierarchy alert archive files present from a previous run, which prevents the subsequent step, load_all_newitem_alert_measures.sh, from completing. This step creates empty files that act as placeholders for the archive files.

Technical Details
The script creates the following zero-byte files needed for the hierarchy alerts:

- $AIPDOMAIN/interface/prod.dat.last
- $AIPDOMAIN/interface/whse.dat.last
- $AIPDOMAIN/interface/loc.dat.last
- $AIPDOMAIN/interface/hspl.dat.last

> **Note:** Other 0-byte alert archive files are created; however, only the files previously listed are required.

Day on Day Processing
This script performs these steps for day on day processing:

1. Remove $AIPDOMAIN/interface/*.last
2. Call UNIX touch to create:
   - $AIPDOMAIN/interface/prod.dat.last
   - $AIPDOMAIN/interface/whse.dat.last
   - $AIPDOMAIN/interface/loc.dat.last
   - $AIPDOMAIN/interface/hspl.dat.last

Appropriate Batch Run (First Time, Daily, or Both)
First Time.

Prerequisites
None.

Restart/Recovery
If create_empty_archive_files.sh fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Generate Batch and Online Alerts

Step Name

create_alerts

Script Call

load_all_newitem_alert_measures.sh

Functional Overview

Each day, alerts must be generated to inform you that new foundation data was loaded into AIP. Once visible in DM Online, the alert prompts you to set up the supply chain for the new data.

The alerts are also used by AIP RPAS batch and DM Oracle batch to perform new hierarchy specific processing, such as Demand Group assignment, Profile assignment, and Store Source assignment.

Technical Details

This step generates AIP Online and AIP RPAS batch alerts. This step compares the new hierarchy data files with the previous set of hierarchy data files (stored as <hierarchy>.dat.last). If it is a first time aip_batch.sh run, the comparison is against empty <hierarchy>.dat.last files created in a previous first-day step, create_empty_archive_files.sh.

Five batch alerts are generated.

Alert Measure	Label
dm0_new	New Stores
dm0_newspl	New Supplier Alert
dm1_new	New Warehouses
dmx_newprd	New Commodities
dmx_newpsz	New Commodity-Pack Sizes

Day on Day Processing

This script performs these steps for day on day processing:

The script generates RPAS batch alerts by using load_one_newitem_alert_measure.sh script.

load_one_newitem_alert_measure.sh compares the old hierarchy, which was saved as hierarchy.dat.last with the new hierarchy that was loaded earlier to obtain new alerts. If one of the files does not exist for comparison, an empty file is generated and then loaded.

The script uses printArray binary to obtain dimensions, hierarchy, starting position, and the width of the alert. It also compares the hierarchy.last.dat with hierarchy.dat file. The difference is saved as measurename.rpl in the input folder of the DM domain. The generated alert measures are then loaded using the loadmeasure binary.

Once the comparison is complete and the alerts are created, the current $AIPDOMAIN/interface/<hierarchy>.dat files are renamed to $AIPDOMAIN/interface/<hierarchy>.dat.last for the next day's alert generation.

This script calls the following script:

- load_one_newitem_alert_measure.sh

Appropriate Batch Run (First Time, Daily, or Both)

- First Time
- Daily

Prerequisites

This step can only run if the load_all_hierarchies.sh script ran successfully.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Check to see if the following measures are flagged correctly based on the new data passed from RMS or AIP Online:
   - dm0_new
   - dm0_newspl
   - dm1_new
   - dmx_newprd
   - dmx_newpsz
4. If any of the previous measures were not flagged correctly, manually update the measures with the correct value:
   - 1 = True
   - 0 = False
5. If necessary, ftp the *.txt files from the external system to $AIPDOMAIN/interface/external.Restart the batch.

Load External System Measure Data into AIP RPAS

The following sections describe how to Load external system measure data into AIP RPAS.

Load Non-RMS External Files

Step Name

load_non_rms_external

Script Call

load_non_rms_external.sh

Functional Overview

This step is used to load non-forecast measures from external sources other than RMS.

Technical Details

This script is a wrapper for load_non_rms_files.sh passed with a command line argument of external. This subscript loads the early arriving non-RMS external system data into the RPAS domain. See "Load Replenishment Inventory Data" on page 5-41 for information about late arriving non-RMS external system data.

Day on Day Processing

This script performs these steps for day on day processing:

1. Create a list of <measure>.ovr file names by reading $AIPDOMAIN/interface/config/external/measdata_from_external.config, and converting all suffixes to ovr.
2. For each of the <measure>.ovr file names in the previous list that refer to existing files in directory $AIPDOMAIN/interface/external:
    1. Move the <measure>.ovr file to directory $AIPDOMAIN/input
    2. Call, in parallel, loadmeasure.sh to load measure <measure> with data from <measure>.ovr.

This script or its sub-script calls the following scripts:

- load_non_rms_files.sh
- loadmeasure.sh (called by load_non_rms_files.sh)

Appropriate Batch Run (First Time, Daily, or Both)

- First Time
- Daily

Prerequisites

This step must not be initiated until successful completion of the following:

- check_process_external_data.sh
- load_all_hierarchies.sh

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. If log files show a measure file did not load properly, but that measure file is already moved to $AIPDOMAIN/input/processed, move the file back to $AIPDOMAIN/interface/external without a timestamp. Files which log files show loaded correctly do not need to be moved back.
3. Correct any identified setup or environment issues.
4. If necessary, restart the batch from this step.

Note: This script may be run in parallel with load_rms_dm_measures.sh and load_onl_data.sh.

Commit Workbooks before Batch Run

The following sections describe how to commit workbooks before batch run.

Auto Commit Workbooks

Step Name
auto_commit_wkbooks_batch

Script Call
workbook_batch.sh COMMIT

Functional Overview
The user-modified workbooks may be configured to automatically commit during the batch. All SRP and WRP workbooks configured for commit at both global and local domain level are committed in this step. This script may also be run from the command line ad hoc throughout the day by administrators or users.

Technical Details
The workbook_batch.sh script is called to process the SRP and WRP workbooks scheduled for automatic commit. The script uses the RPAS utility wbbatch to automatically commit all SRP and WRP workbooks on the workbook commit queue for each global and local domain. The COMMIT action defined by the single parameter is first performed on the global domain workbooks, then in parallel for each of the local domains.

Day on Day Processing
This script performs these steps for day on day processing:

1. Call wbbatch to commit all workbooks on the $AIPDOMAIN global domain commit queue.
2. For each local domain, call wbbatch in parallel to commit all workbooks on each local domain commit queue.

Appropriate Batch Run (First Time, Daily, or Both)
Daily.

Prerequisites
The workday for AIP workbook users must be complete.

Restart/Recovery
If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Perform and Export Data Management Calculations

The following sections describe how to perform and export data management calculations.

Run Initial Load DM Batch

Step Name

run_partial_dm_batch

Script Call

run_partial_dm_batch.sh

Functional Overview

During a first-time run, execute a partial Data Management RPAS Batch process to establish the planning horizons used by AIP RPAS batch, to create and assign warehouse profiles, and to set store source values.

Technical Details

The Data Management binaries dmcreateprf, dmplanhrzn, dmassignprf, and dmcatchup are executed in their entirety.

Day on Day Processing

This script performs these steps for day on day processing:

1. Runs dmcreateprf.sh shell script wrapper to the dmcreateprf AIP binary on the $AIPDOMAIN. This assigns the default Store Order Cycle and all warehouses to automatically created Warehouse Profiles.
2. For each local domain of the RPAS global domain $AIPDOMAIN, runs the dmplanhrzn.sh and dmassignprf.sh shell script wrappers to the dmplanhrzn and dmassignprf AIP binaries. The two scripts are run in series, in the background, for each local domain, such that each local domain is processed in parallel. After all instances are started, the script waits for all instances to complete.
3. For each local domain of the RPAS global domain $AIPDOMAIN, runs the dmcatchup.sh shell script wrapper to the dmcatchup AIP binary. The script is run in the background for each local domain such that each local domain is processed in parallel. After all instances are started, the script waits for all instances to complete.

This Script Calls these Scripts:

- dmcreateprf.sh
- dmplanhrzn.sh
- dmassignprf.sh
- dmcatchup.sh

Appropriate Batch Run (First Time, Daily, or Both)

First Time.

Prerequisites

This step must not be initiated until successful completion of the following:

- Set Implementation Parameters
- Verify and Process Data from External Systems
- Load All Hierarchies

- Load Measure Data
- Load Non-RMS External Data
- Generate Batch and Online Alerts

Restart/Recovery

If this step fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. If there is a problem with the dm1_prfhme measure content, then clear the measure so that the next attempt at overlay load does not have any leftover data from the previous erroneous load attempt.
4. Restart the batch.

Run DM Batch

Step Name

run_dm_batch

Script Call

dmb_master.sh

Functional Overview

DM RPAS batch is a precursor to the Supply Chain Replenishment Planning (SCRP) calculations. It performs a number of setup and maintenance steps that prepare the data for the rest of the batch processes.

The data setup primarily relates to the technical needs of SRP and WRP, known together as SCRP. It also relates to creating various masks or flag measures to ease the processing. Setting up a mask usually involves combining two or more values (for instance; store planning horizon, on-supply dates, and off-supply dates) along with business validation rules in order to produce a single value that can be repeatedly referenced. This keeps the calculation/validation centralized, and it prevents the need for repeating time consuming logic each time the resultant value is needed.

The most notable data setup calculations that are performed include the:

- warehouse schedule
- store release schedule
- store placement schedule

Default Procedures

DM RPAS also performs a number of maintenance and default procedures. These include:

- Warehouse Profile Creation
- New SKU assignment to a warehouse Profile
- Store Source assignment—-Also known as *Catch up*
- Setting warehouse ranging statuses
- Pre-priced status change

Finally, DM RPAS generates alerts that are not related to new hierarchy values. These alerts are exported to DM Online to inform you that various pieces of the supply chain are incomplete or could not be set to the appropriate default settings by DM RPAS. This generation occurs later in the AIP Batch process, however, and not in this step.

Technical Details

At this stage all of the data is loaded into the RPAS global domain $AIPDOMAIN. Now the dmb_master.sh script is called to execute the DM RPAS batch. DM RPAS batch consists of several modules comprised of Korn shell scripts, RPAS rule groups, and C++ binaries. Each module uses some integration (example, RMS, RDF) and intragration (AIP Oracle/Online) inputs and creates outputs, and many times uses as inputs previous DM RPAS batch module outputs. dmb_master.sh is responsible for beginning the critical path processing of DM RPAS batch, so it is called first.

The Korn shell scripts call a number of sub-scripts, which call a number of binaries and execute a number of rule groups in a particular order to calculate new measures. The order of the execution is important because some of the calculated measures are dependent on previously calculated measures. The name of the binary called from these scripts is the same as the name of the script. For example, dmplanhrzn.sh calls the binary dmplanhrzn.

In a later step, the script dmb_master20.sh, which performs non-critical path processing, is executed.

Day on Day Processing

This script performs these steps for day on day processing:

1. dmb_master.sh calls the dmb_master10.sh script, which contains calls to several other DM batch scripts and DM rule groups that perform the critical path processing of Data Management batch in the RPAS domain $AIPDOMAIN. Table 5–5 presents a high level summary of the Data Management modules that are executed along with their functionally descriptive name.

Table 5–5 Summary of the Data Management Modules

Order of Execution	Script / Rule Group	Description
1	dmcreateprf.sh	Assigns Store Order Cycle and Warehouses to automatically created Warehouse Profiles.
2	rulegroup dm_batch_global	Calculate Delivery Pattern Default, Non-release Day Default, Non-order Day Default, Non-delivery Day Default.
3	dmplanhrzn.sh	Calculate Planning Horizons.
4	dmwhsrc.sh	Calculate Warehouse Source.
5	dmassignprf.sh	Attempts to assign new SKUs to warehouse profiles.
6	dmdirsppflg.sh	Direct Supply Point Flag.
7	dmcatchup.sh	Store Source Assignment.
8	horizonFilter.sh dm1_prdalwsts	Filter SKU-Warehouse Allowable Status down to Planning Horizon.
9	dmcmwhal.sh	Ranging Warehouses to SKU-pack sizes.
10	dmstrordpks.sh	Calculate Store Ordering Pack Size.
11	dmstrordunt.sh	Calculate Store Ordering Unit.
12	dmconwhpalmul.sh	Convert Warehouse Pallet Multiple.

Table 5–5 (Cont.) Summary of the Data Management Modules

Order of Execution	Script / Rule Group	Description
13	dmsmlordpksz.sh	Calculate Smallest Ordering Pack Size.
14	dmnerspl.sh	Finds the store's nearest supplier. This supplier will be used for inheriting attributes such as store pallet multiple at the store level.
15	dmdgrspec.sh	Calculate Demand Group Specifier.
16	dmpreprcstchg.sh	Identify Pre-Priced Status Change.
17	dminheritattr.sh	Inherit Attributes.

This script calls the following script:

dmb_master10.sh

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful completion of the following:

- load_non_rms_external.sh
- load_all_newitem_alert_measures.sh
- load_rms_dm_measures.sh
- load_online_measures.sh

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure. This step generates many log files. Navigate through the BSA logging directory structure to find the log file corresponding to the DM batch script or binary which reported an error.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Export DM Data

Step Name

export_dm_data

Script Call

export_dm_data.sh

Required Parameters

firstTime (true | false)

Functional Overview

Certain hierarchy and measure data are required by AIP Online for the users to set up the supply chain. The following data must be made available in a set of flat files that follow an agreed AIP RPAS/AIP Online file format:

- hierarchy data
- measure data calculated in Data Management during the AIP RPAS batch

These are exported from the AIP RPAS domain and bundled for pickup by the client's custom processes to transfer the data to the AIP Online server, for processing by AIP Online's initial batch processes.

Technical Details

The export_dm_data.sh script is called. This script handles exporting the necessary AIP RPAS hierarchy and measure data into flat files formatted for AIP Online.

Day on Day Processing

This script performs these steps for day on day processing:

1. Call the script export_aip_hiers.sh.
   - Copies Supplier and Profile hierarchy files into $AIPDOMAIN/interface/export.
   - Prepares Product, Location and Warehouse hierarchy files for export to AIP on Oracle DB by reducing data files to a prearranged selection of columns.
2. Call the script send_dm_measures_to_online.sh.
   a. This script exports the Data Management measures required for export to AIP Online and copies them to $AIPDOMAIN/interface/export.
   b. It then packages the exported hierarchy and measure data files into the file $AIPDOMAIN/interface/export/dm.tar.Z, a compressed archive that is processed by AIP Online's initial batch step after the client transfers the data from AIP RPAS to AIP Online using a non-AIP process.
   c. Finally this script touches the marker $AIPDOMAIN/interface/export_dm_data indicating that the export has completed. This marker is used by the next day's execution of process_external_data.sh, to determine if the previous day's export was successful.

This Script Calls these Scripts:

- export_aip_hiers.sh
- send_dm_measures_to_online.sh

Appropriate Batch Run (First Time, Daily, or Both)

- First Time
- Daily

Prerequisites

This step must not be initiated until successful completion of the following:

- run_partial_dm_batch.sh (First Time run only)
- dmb_master.sh (Daily run only)

Restart/Recovery

If this step fails, perform the following:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Remove all *.dat and *.ref files from the $AIPDOMAIN/interface/export directory.
4. Remove all dm_* files from the $AIPDOMAIN/interface/export directory.
5. Restart the batch.

Import Schedules from AIP-O into AIP-R

Step Name

check_import_online_schedule

Script Call

check_import_online_schedule.sh

Functional Overview

AIP-O populates the lead time and secondary lead time for all source-destination combinations and needs to be placed by

users into this directory: ${AIPDOMAIN}/interface/import directory. This script creates *.ovr files from the *.dat files. First it clears all the schedule measures and then it then uses these *.ovr files to populate the schedule measures in the AIP-R domain which are then used in further processing.

Technical Details

The check_import_online_schedule.sh is called. Users need to copy the schedule information in the following

directory: ${AIPDOMAIN}/interface/import directory.This data is in the form of *.dat files which are then moved to $AIPDOMAIN/input in the form of *.ovr files (overlays) to be used by loadmeasure to load all the schedule information in the AIP-R domain.

Day on Day Processing

This script performs these steps for day on day processing:

1. The prep_files.sh is invoked with the Schedule_meas_export key that is listed inside $AIPDOMAIN/interface/config/bsa_prep_files.config
2. It then checks whether all the required files present in ${AIPDOMAIN}/interface/config/meas/online_to_rpas_schedule.config are present in $AIPDOMAIN/interface/import/meas or not and exits if any of them is missing.
3. It then moves *.dat files in $AIPDOMAIN/interface/import/meas to *.ovr in $AIPDOMAIN/input for use by loadmeasure.
4. It then clears the schedule measures for full refresh.
5. It then runs loadmeaure to load all schedule measures into the AIP-R domain which will be used for further calculations.

This Script Calls the Following Programs:

- clearArray (AIP binary)
- loadmeasure

This Script Calls these Scripts:

- _check_for_required_files

  A function defined in bsa_check_for_required_files.sh
- prep_files.sh

Appropriate Batch Run (First Time, Daily, or Both)

Daily

Prerequisites

Before running this step, schedule files should be imported from AIP-O into AIP-R.

Restart/Recovery

If this step fails, perform the following:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Check that all the required files listed in ${AIPDOMAIN}/interface/config/meas/online_to_rpas_schedule.config are present in $AIPDOMAIN/interface/import/meas.
4. Restart the batch.

Run Post Schedule Import DM batch

Step Name

dmb_master_post

Script Call

dmb_master_post.sh

Functional Overview

After the schedule information (which contains the lead time and the secondary lead time) has been imported from AIP-O into AIP-R, we need to calculate the schedule dependent measures needed for the batch calculations. This step calculates the review time, DDP, ATP days and the Intra-day nightly prep measures.

Technical Details

The dmb_master_post.sh calls dmb_master_post_local.sh on each of the local domains. In turn, the dmb_master_post_local.sh runs the rule groups that run to calculate the measures needed for further processing of the batch. Since these rule groups require Lead Time, they can only run after importing schedule from AIP-Online.

Day on Day Processing

The dmb_master_post.sh calls dmb_master_post_local.sh which in turn runs the rule groups to populate the necessary measures for the batch calculations.

Table 5–6 presents lists the rule groups that are run along with their functionally descriptive name.

Table 5–6 Rule Groups for Day on Day Processing

Order of Run	Script / Rule Group	Description
1	rulegroup calcAtpMaskDdpRt	Calculate Review Time, DDP and ATP days
2	rulegroup dm_batch_local_p	Calculate all schedule dependent measures
3	rulegroup scrp_pre1	Calculate all supply-chain parameters
4	rulegroup id_nightlyPrep	Calculate Max Lead Time and Replan list for Intra-day

This Script Calls This Script:

dmb_master_post_local

Appropriate Batch Run (First Time, Daily, or Both)

Daily

Prerequisites

This step must not be initiated until successful completion of the following:

check_import_online_schedule.sh

Restart/Recovery

If this step fails, perform the following:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Check that the previous step, check_import_online_schedule.sh ran successfully.
4. Restart the batch.

External Integration Forecast Data Processing and Load into AIP RPAS

The following sections describe the process of external integration forecast data processing and load into AIP RPAS.

Check and Load Forecast Data

Step Name

check_load_forecast_data

Script Call

check_load_forecast_data.sh

Optional Parameters

-h: to display script usage

Functional Overview

AIP manages pulling stock through the supply chain just in time to meet the expected demand. It does not calculate the forecasted store demand, so it must load the calculated demand forecasts for the stores from a demand forecasting system, usually RDF.

Technical Details

This script first checks for files listed as required in the configuration file $AIPDOMAIN/interface/config/forecast/forecastdata_from_external.config. These files must exist in directory $AIPDOMAIN/interface/forecast. Then the script moves both the required and optional files listed in the configuration file to the $AIPDOMAIN/input directory before calling loadmeasure to load each.

Day on Day Processing

This script performs these steps for day on day processing:

1. Verify that all files listed as required in $AIPDOMAIN/interface/config/forecast/forecastdata_from_external.config exist in $AIPDOMAIN/interface/forecast.
2. Create a list of <measure>.ovr file names by reading $AIPDOMAIN/interface/config/forecast/forecastdata_from_external.config, and converting all suffixes to ovr.
3. For each of the <measure>.ovr file names in the previous list that refer to existing files in directory $AIPDOMAIN/interface/external:
    1. Move the <measure>.ovr file to directory $AIPDOMAIN/input.
    2. Call, in parallel, loadmeasure to load measure <measure> with data from <measure>.ovr.

This Script Calls these Scripts:

- _check_for_required_files (function in bsa_check_for_required_files.sh)
- load_non_rms_files.sh
- loadmeasure.sh (called by load_non_rms_files.sh)

Appropriate Batch Run (First Time, Daily, or Both)

Daily only.

Prerequisites

This step must not be initiated until successful completion of the following:

- Forecast data is available from the external system (RDF).

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Ftp the *.txt files from the forecasting system to $AIPDOMAIN/interface/forecast.
4. Restart the batch.

Prepare Replenishment Data and Maintain History

The following sections describe how to prepare replenishment data and maintain history.

Purge and Advance Low-Variability Data

Step Name

purge_low_variability_advance

Script Call

purge_low_variability_advance.sh

Functional Overview

Time-series data measures that hold values that change infrequently in time are stored in a very efficient, specialized encoding that takes advantage of this structure. In order for the AIP calculations to make use of these measures, however, they must be adjusted for the current date and relevant period of history for which AIP calculations are to be performed. This adjustment is a combination of purging information in the past that is too old, while simultaneously advancing key encoded values to the beginning of the relevant time window so that they are not lost. This adjustment must be performed at least once on the day of, and as a precursor to, the running of replenishment calculations.

Technical Details

Some of the data affected by this script is stored at the master level of the domain. Some is stored at the subdomain level. At both the global and the local domain level, the purge process is driven by the aipcmd binary, which processes the command files purgeLowVariabilityAdvanceGlobal.xml and purgeLowVariabilityAdvanceLocal.xml. These XML resource files contain the list of measures on which the purge and advance operation applies at each level. Please see the two XML files, located in $RPAS_HOME/applib/resources, for the lists of measures to which this action applies.

Day on Day Processing

This script performs these steps for day on day processing:

1. Determine the list of local domains under $AIPDOMAIN.
2. Call the aipcmd binary with the purgeLowVariabilityAdvanceGlobal.xml resource file to purge and advance the data stored at the master level.
3. For each local domain, call in parallel the binary aipcmd with the purgeLowVariabilityAdvanceLocal.xml resource file to purge and advance the data stored at the subdomain level.

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful completion of the following steps (script calls).

- workbook_batch.sh COMMIT

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Purge Truncate History

Step Name

purge_truncate_history

Script Call

for_each_local_domain.sh -p purge_truncate_history.sh [DOMAIN]

> **Note:** This complete script call must be made verbatim, including the substring [DOMAIN]. This does not indicate the variable $AIPDOMAIN. Instead, this special string is a token interpreted by for_each_local_domain.sh script as a placeholder for the various local domain paths. for_each_local_domain.sh replaces this token with local domain paths as it calls the purge_truncate_history.sh script for each local domain.

Functional Overview

AIP retains historical data for a few time-series measures that hold value for display on Company Level Inventory Analysis worksheet found on WRP Interactive Evaluation/SRP Interactive Evaluation workbooks. This worksheet displays a configurable length of historical data. The XML Resource file purgeTruncateHistory.xml, located in $RPAS_HOME/applib/resources contains list of base measures used by the previous worksheet along with required historical age in days. The daily AIP batch truncates any excess historical data found in the measures listed, thus maintaining adequate data in the system.

Technical Details

This step's command line script call is a composite of two scripts, for_each_local_domain.sh and purge_truncate_history.sh, which cause the purge truncate history process to be run in parallel across all local domains. At the local domain level, the process is driven by the aipcmd binary, which processes the command file purgeTruncateHistory.xml. This XML resource file contains the list of measures on which the purge truncate history operation applies. Please see file purgeTruncateHistory.xml, located in $RPAS_HOME/applib/resources for the list of measures to which this action applies.

Day on Day Processing

Script for_each_local_domain.sh performs the following steps:

- Determine the list of local domains under $AIPDOMAIN.
- For each local domain, call in parallel the script purge_truncate_history.sh and pass it the local domain path. That path is substituted automatically for the [DOMAIN] token by for_each_local_domain.sh.

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful completion of the following steps (script calls).

- workbook_batch.sh COMMIT

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Copy Sister Stores and Warehouses

Step Name

copy_sister_data

Script Call

for_each_local_domain.sh -p copy_sister_data_local.sh [DOMAIN]

> **Note:** See note for Purge Truncate History regarding the use of for_each_local_domain.sh and its interpretation of [DOMAIN].

Functional Overview

Data to populate measure positions for newly introduced stores and warehouses is often copied to new positions from the positions of so-called sister stores and warehouses. Sister stores and warehouses have similar characteristics and therefore similar position data to the newly introduced positions. Such copying is declaratively configured for a list of affected measures, based on sister-to-new-position map and copy date. When today equals a store or warehouse's copy date, then for all affected measures, data for each new position that corresponds to the copy date is copied from its defined sister position corresponding to the copy date.

Technical Details

This step's command line script call is a composite of two scripts, for_each_local_domain.sh and copy_sister_data_local.sh, which cause the copying of sister store and warehouse data to be performed in parallel across all local domains. At the local domain level, the process is driven by the aipcmd binary, which processes the command files copySisterStore.xml and copySisterWarehouse.xml, both of which use the CopyLikePositions command. The XML resource files contain the lists of measures on which the sister-copying operations apply. Please see files copySisterStore.xml and copySisterWarehouse.xml, located in $RPAS_HOME/applib/resources, for the lists of measures to which this action applies.

Day on Day Processing

Script for_each_local_domain.sh performs the following steps:

1. Determine the list of local domains under $AIPDOMAIN.
2. For each local domain, call in parallel the script copy_sister_data_local.sh and pass it the local domain path. That path is substituted automatically for the [DOMAIN] token by for_each_local_domain.sh.

Appropriate Batch Run (First Time, Daily, or Both)

Daily only.

Prerequisites

This step must not be initiated until successful completion of the following:

- workbook_batch.sh COMMIT
- load_onl_data.sh

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

External Integration Inventory Data Processing and Load into AIP RPAS

The following sections describe the process of external integration inventory data processing and load into AIP RPAS.

Verify and Process Inventory Data from External System

Step Name

check_process_inventory_data

Script Call

check_process_inventory_data.sh

Optional Parameters

-h: to display script usage.

Functional Overview

RMS is an enterprise solution that provides merchandise hierarchy and organizational setup and maintenance. This data, together with non-RMS external system data, becomes the foundation for the AIP supply chain replenishment. The data is processed as a set of flat files that follow an agreed AIP file format. If the client does not use RMS, but rather some other external system, this data is still required in the same format. AIP validates all of the files to ensure that the required data is present, and massages the files to produce the input required for subsequent steps of the AIP RPAS batch. AIP divides the integration data into two categories: early files and late files. The batch schedule for AIP may aim to maximize the available time by moving as much workload off the critical path as possible.

For this reason the static data is extracted from RMS, or similar system, before the dynamic transaction data is available. This allows the hierarchies to be loaded into AIP

and the DM batch run before the dynamic data is extracted. This step processes the early, static data.

Technical Details

The dynamic measure data are listed in the following table. For AIP RPAS batch to properly process external system data, all of the flat files must follow a particular format, which is explained in the *Oracle Retail Advanced Inventory Planning Implementation Guide*. The flat files must have a *.txt extension.

File Name	Description
sr0_curinv_[1..n].txt	Store Current Inventory
sr0_it_.txt	Store In Transits
sr0_oo_.txt	Store On Orders
sr0_prdlfe.txt	Store Product Life
wr1_curinv.txt	Warehouse Current Inventory
wr1_it_.txt	Warehouse In Transits
wr1_oo_.txt	Warehouse On Orders
wr1_aiw.txt	Warehouse Allocations in the Well
wr1_tiw.txt	Warehouse Transfers in the Well

Day on Day Processing

This script performs these steps for day on day processing:

1. Verify that all required files have been downloaded. Failure to download any of the required files results in an error and termination of the batch.
2. If store current inventory sr0_curinv data contains more than one partition, like sr0_curinv_1.txt, sr0_curinv_2.txt., …, sr0_curinv_n.txt, merge all of the partitioned files. Rename single or consolidated data file to sr0_curinvinit.txt.
3. Rename wr1_curinv.txt to wr1_curinvinit.txt.
4. Prefix the following measure data with the stocking point prefixes, using construct_ntier_measuredata.ksh:
   - sr0_curinvinit.txt
   - sr0_it_.txt
   - sr0_oo_.txt
   - wr1_curinvinit.txt
   - wr1_it_.txt
   - wr1_oo_.txt
   - wr1_aiw.txt
   - wr1_tiw.txt
5. Using interutil binary, convert from RMS SKU to AIP SKU all RMS-sourced inventory measure data that has been configured to arrive late, and therefore is present to be processed by this script (as opposed to arriving early, and therefore is processed by check_process_external_data.sh). See the earlier script and the *Oracle Retail Advanced Inventory Planning Implementation Guide* for details on this

configuring. Note the AIP configuration is to have all dynamic data listed in the previous table arrive as late.

This Script Calls these Scripts:

- _check_for_required_files (defined in bsa_check_for_required_files.sh)
- process_inventory_data.sh

Appropriate Batch Run (First Time, Daily, or Both)

Daily only.

Prerequisites

This step must not be initiated until successful completion of the following:

- All data files from the previous table should be copied by the client's batch scheduler into the $AIPDOMAIN/interface/rms directory. Some of the data is required, and some is optional. Reference the $AIPDOMAIN/interface/config/external/latefiles.config for requirements.

Restart/Recovery

If the process_inventory_data.sh script failed, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Re-copy or re-FTP all required files listed in $AIPDOMAIN/interface/config/external/latefiles.config into the $AIPDOMAIN/interface/rms directory. The new copies should overwrite the existing files in this directory.
4. Restart the batch.

Notes

This step can be run in parallel with Export DM and OM Online and all aip_batch.sh steps up to scrp.sh.

Load Replenishment Inventory Data

Step Name

load_replenishment_data

Script Call

load_replenishment_measures.sh

Functional Overview

This script preprocesses and loads into the RPAS domain all store and warehouse replenishment measure data that come from RMS or other external inventory system.

Technical Details

This script loads into $AIPDOMAIN all Store and Warehouse measure data in $AIPDOMAIN/interface/rms. This step also loads the late arriving non-RMS external data.

Day on Day Processing

This script performs these steps for day on day processing:

1. Call loadmeasure.sh (a script wrapper for loadmeasure RPAS binary) on the measure data files corresponding to the Replenishment measure data received from the inventory system (example, RMS). The measures loaded are listed in the following configuration files:
   - $AIPDOMAIN/interface/config/rms/srp_rms_measures.config
   - $AIPDOMAIN/interface/config/rms/wrp_rms_measures.config

   Note all measures are loaded as full replacement (.rpl). This is not intended to be configurable due to functional requirements.
2. Call load_non_rms_files.sh on the data files corresponding to the late arriving non-RMS external measure data files in the $AIPDOMAIN/interface/external.

This Script Calls these Scripts:

- loadmeasure.sh
- load_non_rms_files.sh

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful completion of the following:

- All steps through the load_all_hierarchies.sh script.
- This step can be run in parallel with Load Measures with AIP Online Data.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Ensure that the *.txt files are transferred through FTP by client scheduled process from RMS to the correct location.
4. Ensure all RMS file are present.
5. Ensure all files are formatted correctly.
6. Ensure that the Load All Hierarchies step ran correctly, and all of the hierarchies' values received from RMS are loaded correctly.
7. Restart the batch.

Calculate and Export Replenishment Plan

The following sections describe how to calculate and export the replenishment plan.

Run Replenishment

Step Name

run_replenishment

Script Call

scrp.sh

Functional Overview

Replenishment and Reconciliation is part of the AIP batch run which generates a Receipt Plan for Warehouses and Stores across the length of the horizon. The final output from this batch process is 1) Constrained Receipt Plan for Warehouses and Stores across the Fixed Period. 2) Unconstrained Receipt Plan for Warehouses and Stores across the post Fixed period up to the end of the horizon.

Technical Details

This process is run by running the master script scrp.sh. This script initiates the batch run on global and local domains by first calling the scrp_global.sh on the global domain and then simultaneously calling the scrp_local.sh on all local domains. The global and local scripts internally invoke a set of rule groups according to the sequence subsequently described. Each rule group is responsible for performing a specific step in the supply chain replenishment planning process.

Table 5–7 lists the scripts used for the Replenishment/Reconciliation batch run.

Table 5–7 Scripts Used for the Replenishment/reconciliation Batch Run

Script Name	Description
scrp.sh	This script calls scrp_global.sh first on the global domain and then calls scrp_local.sh on all local domains
scrp_global.sh	Runs the batch on the global domain
scrp_local.sh	Runs the batch on local domains

Table 5–8 lists all of the rule groups involved in the AIP Replenishment and Reconciliation batch process.

Table 5–8 Rule Groups Involved In the AIP Replenishment & Reconciliation Batch Process

Rule Group	Description
Global Rule Groups	This rule group is called by the scrp_global.sh script.
scrp_pre_glb	Generates Forecast percentages, and sets store singles flag.
Local Rule Groups	The following rule groups are called by scrp_local.sh script.
scrp_StrDynamic	Runs Store Dynamic Safety Stock Statistical Calculation.
scrp_WhDynamic	Runs Warehouse Dynamic Safety Stock Statistical Calculation.
scrp_pre2	Runs all the pre replenishment steps for AIP.
scrp_replPH	Runs the replenishment batch run within the planning horizon and generates an unconstrained receipt plan.
scrp_invCpPH	Restricts the unconstrained plan of stores within the planning horizon to inventory caps set by user on workbook. The demand forecasted on the warehouse remains uncapped or the True Demand.
scrp_prReconcile	Runs pre reconciliation steps for AIP.
scrp_reconcile	Runs the shortfall, SPQ and stockless to warehouse reconciliation when the demand is greater than the available inventory at the source and generates a constrained receipt plan.

Table 5–8 (Cont.) Rule Groups Involved In the AIP Replenishment & Reconciliation Batch Process

Rule Group	Description
scrp_reconcSub	Runs the substitution reconciliation and substitutes alternate allowed pack sizes when there is still some unmet demand from destinations.
scrp_pushToStore	Runs the stockless reconciliation at store destinations for stockless items and pushes the excess quantity to the stores.
scrp_invCpRec	Checks if the capped plan was reduced further due to shortfall reconciliation and adjusts the warehouse demand accordingly.
scrp_replPstFxd	Runs the replenishment batch run post fixed period and generates an unconstrained receipt plan for post fixed period.
scrp_invCpPstFxd	Restricts the post fixed period unconstrained plan of stores to inventory caps set by user on workbook.
WipConvOrders	Used to map warehouse to warehouse chamber.

Day on Day Processing

scrp.sh script is the master script to be called once on the global domain. It calls scrp_global.sh on the global domain first and then calls scrp_local.sh on all local domains.

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful completion of the following:

- load_replenishment_measures.sh
- check_load_forecast_data.sh
- dmb_master.sh
- for_each_local_domain.sh -p copy_sister_data_local.sh [DOMAIN]

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Run dmb_master.sh.
4. Restart the batch.

Export Supply Chain Replenishment Data

Step Name

export_replenishment_data

Script Call

for_each_local_domain.sh -p export_scrp_inter_meas_local.sh [DOMAIN]

Note: See note for Purge and Advance Low-Variability Data regarding the use of for_each_local_domain.sh and its interpretation of [DOMAIN].

Functional Overview

The replenishment plan's planned orders from suppliers and transfers from warehouses must be exported from populated measures into files suitable for interfacing with AIP Online. In this batch step, the exports are performed independently in each local domain, resulting in each having its own copy of the following export files placed in the local domain output directory:

Script Name	Description
strsplrord.dat	Supplier to store orders for release today through the planning horizon.
strsplrord.dat1	Contingency supplier to store orders for release tomorrow only.
strwhord.dat	Warehouse to store transfers for release today.
strwhord.dat1	Contingency warehouse to store orders for release tomorrow only.
vendor_to_wh_order.dat	Supplier to warehouse orders for release today through the planning horizon.
wh_to_wh_transfer.dat	Warehouse to warehouse transfers for release today through the planning horizon.

Technical Details

This step's command line script call is a composite of two scripts, for_each_local_domain.sh and export_scrp_inter_meas_local.sh, which together cause the exporting of order and transfer data to be performed in parallel across all local domains. At the local domain level, the process is driven by the aipcmd binary, which processes the XML resource command files to produce the parenthesized output files:

- exportPlanStrSplrOrd.xml (strsplrord.dat)
- exportPlanStrSplrOrd1.xml (strsplrord.dat1)
- exportPlanStrWhOrd.xml (strwhord.dat)
- exportPlanStrWhOrd1.xml (strwhord.dat1)
- exportPlanWhOrd.xml (vendor_to_wh_order.dat)
- exportPlanWhXfer.xml (wh_to_wh_transfer.dat)

These XML resource files all make use of the ExportPlan command to perform the specialized data exports.

Day on Day Processing

Script for_each_local_domain.sh performs the following steps:

1. Determine the list of local domains under $AIPDOMAIN.
2. For each local domain, call in parallel the script export_scrp_inter_meas_local.sh and pass it the local domain path. That path is substituted automatically for the [DOMAIN] token by for_each_local_domain.sh.

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful completion of the following:

- scrp.sh

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Package Supply Chain Replenishment Data

Step Name

send_replenishment_to_online

Script Call

send_scrp_measures_to_online.sh

Functional Overview

Planned orders and transfers, exported separately per local domain in the Export Replenishment Data step, are combined, tarred, and compressed for interfacing with AIP Online.

Technical Details

In this batch step, the filenames listed in the global domain's interface/config/meas/scrp_export.config file are located in each local domain's output directory and concatenated into like-named files in the global domain's interface/export directory.

Day on Day Processing

This script performs these steps for day on day processing:

1. Verify the existence and writeability of the global domain's interface/export directory, the destination of the concatenated export files.
2. For each of the files listed in the global domain's interface/config/meas/scrp_export.config file, delete any existing instance in the global domain's interface/export directory, then iterate over the list of local domains, concatenating each instance of the file found in the local domain's output directory into the global interface file of the same name in global domain's interface/export directory.
3. Compress each global domain export file, package them all into a single tar file, then compress the tar file into a single file: $AIPDOMAINinterface/export/srp.tar.Z.

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful completion of the following:

- for_each_local_domain.sh -p export_scrp_inter_meas_local.sh [DOMAIN]

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Import Cross Dock Constraint Receipt Plan (DCRP) Output from AIP-O into AIP-R

Step Name

post_dcrp_import

Script Call

post_dcrp_import.sh

Functional Overview

AIP-O populates the Cross Dock Constraint Receipt Plan(DCRP) measures and places them in the form of a zip file in the ${AIPDOMAIN}/interface/import directory. This script creates *.ovr files from the *.dat files contained in the zip file. First it clears all the DCRP output measures and then it then uses these *.ovr files to populate these measures in the AIP-R domain which are then used in further processing.

Technical Details

The post_dcrp_import.sh is called. AIP-O places all the DCRP output information in the form of a zip file, dcrp.tar.Z, in the ${AIPDOMAIN}/interface/import directory. This script then unpacks the data contained in dcrp.tar.Z to $AIPDOMAIN/interface/import/meas directory. This data is in the form of *.dat files which are then moved to $AIPDOMAIN/input in the form of *.ovr files (overlays) to be used by loadmeasure to load all the DCRP outputs in the AIP-R domain.

Day on Day Processing

This script performs these steps for day on day processing:

1. The prep_files.sh is invoked with the DCRP_meas_export key that is listed inside $AIPDOMAIN/interface/config/bsa_prep_files.config

   This script processes the file dcrp.tar.Z contained in ${AIPDOMAIN}/interface/import and unpacks the data contained within to the $AIPDOMAIN/interface/import/meas directory.The data unpacked overwrites any existing files in the directory.
2. It then checks whether all the files present in ${AIPDOMAIN}/interface/config/meas/post_dcrp_import.config are present in $AIPDOMAIN/interface/import/meas or not and exits if any of them is missing.
3. It then moves *.dat files in $AIPDOMAIN/interface/import/meas to *.ovr in $AIPDOMAIN/input for use by loadmeasure.
4. It then clears the DCRP measures for full refresh.

5. It then runs loadmeasure to load all DCRP output measures into the AIP-R domain which will be used for further calculations.

This Script Calls the Following Programs:

- clearArray (AIP binary)
- loadmeasure

This Script Calls these Scripts:

- _verify_file_exists

  A function defined in bsa_verify.sh

- prep_files.sh

Appropriate Batch Run (First Time, Daily, or Both)

Daily

Prerequisites

Before running this step, dcrp.tar.Z should be imported by AIP-O into AIP-R.

Restart/Recovery

If this step fails, perform the following:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Ensure that dcrp.tar.Z is present in ${AIPDOMAIN}/interface/import directory.
4. Check that all the files listed in ${AIPDOMAIN}/interface/config/meas/post_dcrp_import.config are present in $AIPDOMAIN/interface/import/meas.
5. Restart the batch.

Load the Scaled Replenishment Plan

The following sections describe how to load the scaled replenishment plan.

Run Post Supplier and Container Scaling Import

Step Name

post_ocs_import

Script Call

post_ocs_import.sh

Functional Overview

Modifications made to the warehouse plan by the Scaling module are imported so that the future projections of inventory and ordering, which are displayed in workbooks and reflected in alert calculations, are based on the final planned order quantities.

This step can be excluded from the batch if the business does not use the Scaling module within AIP. This step must not run if the corresponding AIP Oracle database extract is not run to generate the data files.

Technical Details

None.

Day on Day Processing

This script performs these steps for day on day processing:

1. prep_files.sh is called to process the AIP Oracle scaling export. The prep_files.sh script is invoked with the key OCS_meas_export listed inside $AIPDOMAIN/interface/config/bsa_prep_files.config. This processes the file ocs.tar.Z and unpacks the data contained within to the $AIPDOMAIN/interface/import/meas directory.
   - Before the archive is unpacked, a backup of the file is created in the $BSA_ARCHIVE_DIR directory. The backup filename is ocs.tar.Z.<timestamp>.
   - After the archive is unpacked, the archive file is removed from $AIPDOMAIN/interface/import.
   - The data unpacked overwrites existing files in the directory.
2. Verify that all files have been downloaded and unpacked. Failure to download any of the files results in an error and termination of the batch. Required files are found in post_ocs_import.config.
3. If the SPQ Date Type measure holds the value for *Ship Date* then the delivery date in the file containing Executed SPQ (SPEQ) quantities is converted to TODAY, the ship date. The file extension is then changed to .inc (incremental) to cause the values loaded into the same position to be added together. This need can arise when the SPQs are applied to the Ship week and multiple delivery dates have the same release (ship) date.
4. The measures to be loaded are cleared and then loaded with data from the files.

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful ftp or copy of the ocs.tar.Z file.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues. If necessary, copy and rename the archived version of the ocs.tar.Z file from the archive directory into the $AIPDOMAIN/interface/import directory.
3. Restart the batch.

Perform Post-Replenishment Calculations

The following sections describe how to perform post-replenishment calculations.

Run Replenishment Post-Processing

Step Name

run_replenishment_post_processing

Script Call

for_each_local_domain.sh -p scrp_post_local.sh [DOMAIN]

> **Note:** See note for Purge and Advance Low-Variability Data regarding the use of for_each_local_domain.sh and its interpretation of [DOMAIN].

Functional Overview

Replenishment post processing generates Safety Stock, Projected Inventory, Receipt Point, and Receive Up To Level values across the horizon. These values are calculated on the fly during the Replenishment and Reconciliation stage but are not written to a measure as the demand and inventory picture changes across the Fixed Period. Once the constrained receipt plan is generated, the correct demand and inventory levels are known for the post processing to generate the previous values across the horizon.

Also, when order and container scaling is implemented, it updates Receipt Plan, Forecasted Receipts, Demand Output, and Total Forecast Demand.

Technical Details

Table 5–9 lists the scripts used for the Replenishment Post Processing batch run.

Table 5–9 Scripts Used for the Replenishment Post Processing Batch Run

Script Name	Description
scrp_post_local.sh	This script calls scrp_post rule group on the local domains which actually does the post processing.

Table 5–10 gives a list of all the rule groups involved in the AIP Post Replenishment & Reconciliation batch process.

Table 5–10 Rule Groups Used In the AIP Post Replenishment & Reconciliation Batch Process

Rule Groups	Description
Local Rule Groups	The following rule groups are called by scrp_post_local.sh script.
scrp_post_ocs1	Copies replenishment data to the pre-scaling measures.
scrp_post_ocs2	Adds Scaling adjustments to the replenishment data.
scrp_post	Performs the post processing on local domains and generates inputs for calculating historical measures and alerts.
scrp_histmaint	Preserves data for certain measures for historical purposes which are especially useful for alerts.
scrp_cleartemp	Clears some of the stored inputs calculated in scrp_post now that its data was preserved in scrp_histmaint.

Day on Day Processing

Script for_each_local_domain.sh performs the following steps:

1. Determine the list of local domains under $AIPDOMAIN.
2. For each local domain, call in parallel the script scrp_post_local.sh script sh and pass it the local domain path. That path is substituted automatically for the [DOMAIN] token by for_each_local_domain.sh.

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful completion of the following:

- scrp.sh
- post_ocs_import.sh (if implementing order and container scaling)

Restart/Recovery

This step must not be initiated until successful completion of the following:

- scrp.sh
- post_ocs_import.sh (if implementing order and container scaling)

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. If order and container scaling is not implemented: Restart the batch.
4. If order and container scaling is implemented:
   - Determine the local domains that had an issue.
   - Determine which of the previous rule groups ran successfully.
   - If none ran successfully: restart the batch.
   - If some ran successfully: Re-run manually the rule groups that did not run successfully. Then restart the batch.

Calculate and Export Data Management Alerts

The following sections describe how to calculate and export data management alerts.

Run Non-critical Data Management Alerts

Step Name

run_dm_alerts

Script Call

dmb_master_alerts.sh

Required Parameters

firstTime (true | false)

Functional Overview

DM RPAS generates alerts that are not related to new hierarchy values. These alerts are exported to DM Online to inform you that various pieces of the supply chain are incomplete or could not be defaulted by DM RPAS.

Technical Details

Now the dmb_master_alerts.sh script is called to execute the DM RPAS batch, non-critical path, including the alerts generation. As before, this process is comprised of several modules that use Korn shell scripts and C++ binaries in a particular order to calculate new measures. The order of the execution is important because some of the calculated measures are dependent on previously calculated measures.

Day on Day Processing

dmb_master_alerts.sh calls the dmb_master20.sh script, which calls dmb_master21.sh on all local domains, which contains calls (through subscripts) to several other DM batch scripts that perform the non-critical path processing of Data Management batch in the RPAS domain $AIPDOMAIN. Table 5–11 presents a high level summary of the Data Management modules that are executed along with their functionally descriptive name.

Table 5–11 Summary of the Data Management Modules

Order of Execution	Script / Operation	Description
1	mace call	Calculate Warehouse Demand
2	dmcmdstralerts.sh	Commodity/Store Alerts
3	horizonFilter.sh	Prepare time-boxed Product Allowable Status Measure for Use by Missing Data Alert
4	dmmsgdatalert.sh	Missing Data Alert
5	dmnasprfalert.sh	Not Assigned to Profile Alert
6	dmdscalert.sh	Discontinuation Alert

This Script Calls these Scripts:

- xmace
- dmb_master20.sh

Appropriate Batch Run (First Time, Daily, or Both)

- First Time
- Daily

Prerequisites

This step must not be initiated until successful completion of the following:

- run_partial_dm_batch.sh (First Time run only)
- dmb_master.sh (Daily run only)

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Export DM Alerts

Step Name
export_dm_alerts

Script Call
export_dm_alerts.sh

Required Parameters
FirstTime (true | false)

Functional Overview
The Data Management Alert measures calculated by the processes of dmb_master20.sh are required for import into AIP Online. These are exported from the AIP RPAS domain and bundled for pickup by AIP Online's initial batch processes.

Technical Details
The export_dm_alerts.sh script is called. This script handles exporting the necessary AIP RPAS alert data into flat files formatted for AIP Online.

Day on Day Processing
This script performs these steps for day on day processing:

1. Call the script export_dm_inter_meas.sh to export the intragration alert data from the calculated alert measures.
2. Package the exported alert measure data files into the file $AIPDOMAIN/interface/export/dm_alerts.tar.Z, a compressed archive that is imported by AIP Online's initial batch step.

This script calls the following script:
export_dm_inter_meas.sh

Appropriate Batch Run (First Time, Daily, or Both)
- First Time
- Daily

Prerequisites
This step must not be initiated until successful completion of the following:

- dmb_master_alerts.sh

Restart/Recovery
If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.

3. Restart the batch.

Compute Replenishment Alerts

The following sections describe how to compute replenishment alerts.

Run SRP Item Alerts

Step Name

run_srp_alerts

Script Call

scrp_srp_alerts.sh

Functional Overview

All of the SRP alerts for out-of-stocks and excessive planned orders must be generated by AIP RPAS batch. These alerts allow the retailer to identify potential supply chain problems before they happen so that potential stock-outs and excess inventory problems can be prevented or reduced.

Technical Details

This step runs the SRP Alert calculations after the batch jobs have finished. It is recommended that this part of the system run off the critical path.

During the domain build the following SRP alerts measures are registered in the AIP domain:

Alert Name	Description
sr0_art__1	Store Alert 1: Large Consecutive Out of Stocks
sr0_art__2	Store Alert 2: Large Out of Stocks Last Night
sr0_art__3	Store Alert 3: Single Store Availability Problems
sr0_art__4	Store Alert 4: High Projected Out of Stock
sr0_art__5	Store Alert 5: Large Non-Consecutive Out of Stocks
sr0_art__6	Store Alert 6: Day on Day Repeat Out of Stocks
sr0_art__7	Store Alert 7: High Projected Low Stocks
sr0_art__8	Store Alert 8: High Planned Orders
sr0_art__9	Store Alert 9: RDF Detail Alert
sr0_art__10	Store Alert 10: No Like SKU Found
sr0_arthstsum	Historic Availability Alert
sr0_artprjsum	Projected Availability Alerts
sr0_ovrstkalt	Overstock Alert
sr0_hrdart	High Repeated Dissipation
sr0_hdyart	High Dissipation Yesterday

Day on Day Processing

This script performs these steps for day on day processing:

1. Execute scrp_glbMsk rule group.
2. In each local domain, run in parallel these two rule groups:
   - scrp_preRegAlert
   - scrp_srpAlerts

This script calls the following script:
xmace

Appropriate Batch Run (First Time, Daily, or Both)
Daily.

Prerequisites
This step must not be initiated until successful completion of the following:

- scrp.sh
- for_each_local_domain.sh -p scrp_post_local.sh [DOMAIN]

Restart/Recovery
If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Notes
This step calculates the basics on which SRP alerts are based. Later, the replenishment_alerts_post_processing.sh runs the RPAS utility alertmgr to find all alerts registered in the $AIPDOMAIN domain.

Run WRP Item Alerts

Step Name
run_wrp_item_alerts

Script Call
wrp_item_alerts.sh

Functional Overview
The WRP alerts for unmet demand and warehouse capacity issues are generated by AIP RPAS batch. These alerts allow the retailer to identify potential supply chain problems before they happen so that potential stock-outs and excess inventory problems can be prevented or reduced.

Technical Details
This step runs the WRP Alerts after the batch jobs have finished. It recommended that this part of the system run off the critical path.

During the domain build the following WRP Item alerts measures are registered in the AIP domain:

Alert Name	Description
IpSlsCrdAltV	Sales Credit Alert
IpDmdCrdAltV	Demand Credit Alert
IpOvrStkAltV	Overstock Alert
IpRdfAltV	RDF Alert

Day on Day Processing

This script performs these steps for day on day processing:

1. Determine the list of local domains under $AIPDOMAIN.
2. For each local domain, run in parallel the WrpItem_alert rule group.

This script calls the following script:

xmace

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful completion of the following:

- scrp.sh
- for_each_local_domain.sh -p scrp_post_local.sh [DOMAIN]

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Note: This step calculates the basics on which WRP item alerts are based. Later, the replenishment_alerts_post_processing.sh runs the RPAS utility alertmgr to find all alerts registered in the $AIPDOMAIN domain.

Run WRP Network Alerts

Step Name

run_wrp_network_alerts

Script Call

wrp_network_alerts.sh

Functional Overview

The WRP Network alerts notify the user of various product flow scenarios across user-defined network of SKUs. These alerts highlight potential supply chain problems

before they happen, so that potential stock-outs and excess inventory problems can be prevented or reduced.

Technical Details

This step runs the WRP Network Alerts after the batch jobs have finished. It is recommended that this part of the system run after the critical path.

During the domain build the following WRP Network alerts measures are registered in the AIP domain.

Alert Name	Description
IpStkCvrAltV	Stk Cvr Alert (Act In vs. Act Out)
IpIbDODAltV	Inbound Day-on-Day Change Alert
IpDstObCpcAltV	Outbound Distribution Capacity Alert
IpObDODAltV	Outbound Day-on-Day Change Alert
IpScDODAltV	Stock Cover Day-on-Day Change Alert
IpHldCpcAltV	Warehouse Holding Capacity Alert

Day on Day Processing

This script performs these steps for day on day processing:

1. Clear all network alert measures in $AIPDOMAIN.
2. Determine the list of local domains under $AIPDOMAIN.
3. For each local domain, run in parallel the WrpNwAlertLocal rule group.
4. Execute the WrpNwAlertGlobal on the global domain $AIPDOMAIN.

This script calls the following script:

xmace

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until successful completion of the following:

- scrp_srp_alerts.sh
- wrp_item_alerts.sh

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Note: This script should not be run consecutively on the same day as it affects the day-on-day change alerts.

Replenishment Alerts Post Processing

Step Name
run_replenishment_alerts_post_processing

Script Call
replenishment_alerts_post_processing.sh

Functional Overview
The WRP Network alerts notify the user of various product flow scenarios across user-defined network of SKUs. These alerts highlight potential supply chain problems before they happen, so that potential stock-outs and excess inventory problems can be prevented or reduced.

In this step, the history is updated to contain the current day's run.

Technical Details
This step finds the WRP Network Alerts after the batch jobs have finished. It recommended that this part of the system run after the critical path.

Day on Day Processing
This script performs these steps for day on day processing:

1. Call alertmgr to find all generated alerts on $AIPDOMAIN.
2. Execute the WrpNwAlertGlobal2 on the global domain $AIPDOMAIN.

This script calls the following script:
xmace

Appropriate Batch Run (First Time, Daily, or Both)
Daily.

Prerequisites
This step must not be initiated until successful completion of the following:

- wrp_network_alerts.sh

Restart/Recovery
If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

> **Note:** This script should not be run consecutively on the same day as it affects the day-on-day change alerts.

Building Workbooks after Batch Run

The following sections describe how to build workbooks after a batch run.

Auto Build Workbooks

Step Name
auto_build_wkbooks_batch

Script Call
workbook_batch.sh BUILD

Functional Overview
Many of the SRP and WRP workbooks can be configured by the system administrator to be automatically built each night as part of the AIP RPAS batch run. This allows you to enter the worksheets directly without going through the workbook creation wizard. All SRP and WRP workbooks configured for commit at both global and local domain level are committed in this step. This script may also be run from the command line ad hoc throughout the day by administrators or users.

Technical Details
This step uses the RPAS utility wbbatch to automatically build SRP and WRP workbooks. In order to successfully auto-build workbooks, they must be configured through the RPAS client. Refer to the RPAS Administration Guide for information on automatically building workbooks.

Day on Day Processing
This script performs these steps for day on day processing:

1. Call wbbatch to build all workbooks on the $AIPDOMAIN global domain build queue.
2. For each local domain, call wbbatch in parallel to build all workbooks on each local domain build queue.

Appropriate Batch Run (First Time, Daily, or Both)
Daily.

Prerequisites
This step must not be initiated until the following steps (scripts) are successfully completed.

- scrp_srp_alerts.sh
- wrp_item_alerts.sh
- wrp_network_alerts.sh
- replenishment_alerts_post_processing.sh

Restart/Recovery
If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Run Calculations for OBIEE Reports

The following sections describe how to run calculations for OBIEE reports.

Calculate Data for OBIEE Reports

Step Name

run_reports_calculation

Script Call

run_OBIEE_reports.sh

Functional Overview

The Integrated Inventory Planning (IIP) OBIEE Reports direct the Inventory Analyst's attention to problematic situations summarized by SKU and by warehouse. The reports help direct the user's analysis starting with a summarized view of the problem and providing greater detail at each reporting level. This helps the user to not only identify potential issues quickly but also quickly analyze the size and severity of the issue to prioritize the resolution.

The IIP Reports will integrate data from the following Oracle Retail applications: Advanced Inventory Planning (AIP), Retail Demand Forecasting (RDF), Replenishment Optimization (RO), and Retail Merchandising System (RMS). This step runs the AIP calculations required to support the OBIEE reports desired for IIP.

Technical Details

This step uses the RPAS mace utility to run several rule groups which calculate the reporting measures.

Day on Day Processing

This script performs these steps for day on day processing:

1. Run xmace wrapper to mace for the following rule groups:
   - Common Rules
   - Capacity Report
   - New SKU Report
   - Discontinued SKU Report
   - Alert Dashboard

This script calls the following script:

xmace

Appropriate Batch Run (First Time, Daily, or Both)

Daily.

Prerequisites

This step must not be initiated until the following steps are successfully completed.

- run_replenishment_alerts_post_processing

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Run Dashboard

The following sections describe how to run the AIP dashboard.

Calculate AIP Dashboard Data

Step Name

run_dashboard

Script Call

run_dashboard.sh

Functional Overview

The AIP Dashboard feature is part of AIP Usability enhancements for the RPAS portion of the AIP application. The AIP Dashboard is a view of information relevant to a user, available to them in a high level view, without accessing a workbook first.The Dashboard is intended to provide quick visibility to important data for items of interest to users through the features Product Watchlist, Location Watchlist and Orders about to be executed.The AIP Dashboard is accessed by AIP RPAS users from the Fusion Client after authentication.

Technical Details

The run_dashboard.sh script is called which runs the dashboard rule group on each of the local domains which populates the measures which are used in AIP dashboard.

Day on Day Processing

This script runs the dashboard rule group on the local domains and populates the following dashboard-only measures:

- IpPlnRcpV
- sr0_pckssday
- sr0_pcksswk
- sr0_pckrutlday
- sr0_pckrutlwk
- sr0_pckrpday
- sr0_pckrpwk
- sr0_pckdmdwk
- sr0_pckdmdday
- sr0_strttlplnrcpwk
- sr0_strprjinvwk

- sr0_strprsstkwk
- IpWhSSWkV
- IpWhRUTLWkV
- IpWhRcpPtWkV
- IpWhPrjInvWkV
- IpTtlDmdV
- IpTtlDmdWkV
- IpWhPlnRcpWkV
- IpWhPlnRcpTmpO
- IpWhSSTmpO
- IpWhPlnRcpV
- IpWhPrjInvV
- IpPlnRcpRelV
- IpPlnExpRcpDelV
- IpWhRcpPtV
- IpWhRcpPtTmpO
- IpWhRUTLV
- IpWhRUTLTmpO
- IpWhSSV
- wr1_prjinv
- sr0_prjinv
- sr0_strprjinv
- sr0_rcppln
- sr0_strttlplnrcp
- sr0_strprsstk
- IpExpRcpSrcV

This Script Calls this Script:
xmace

Appropriate Batch Run (First Time, Daily, or Both)
Daily

Prerequisites
This step must not be initiated until successful completion of the following:

run_OBIEE_reports.sh

Restart/Recovery
If this step fails, perform the following:

1. Examine the log files to determine the cause of the failure.

2. Correct any identified setup or environment issues.
3. Restart the batch.

6

AIP Java/Oracle Batch Process Flow

The top level control scripts (cron_export.sh, cron_import.sh, cron_import_order.sh, pre_scale.sh, smooth_order.sh, scale_order.sh, post_scale.sh, cron_release.sh, cron_post_release.sh, and cron_purge.sh) are provided with the AIP installation. They may be used to run the AIP Oracle batch process from a UNIX scheduler. There are many other internal scripts called by these scripts that may have multi-threaded process calls; however, all the scripts called directly by cron_export.sh, cron_import.sh, cron_import_order.sh, cron_release.sh, and cron_purge.sh are run inline.

Table 6–1 describes the three categories of batch scripts based on their execution frequency.

Table 6–1 Batch Script Categories

Batch Scripts	Description
Intra-day Batch	These scripts are used for Intra-day batch processes and can be called multiple times in a single day. They are executed once for each Intra-day batch run. The Intra-day batch scripts currently apply to both into-store and into-warehouse purchase orders and transfers.
Overnight/Regular Batch	These scripts are used for regular overnight batch run and are executed once-a-day in normal mode. Some of these can be used in Intra-day mode also when executed with optional release wave parameter. These scripts apply to both into-store and into-warehouse type purchase orders and transfers.
Weekly/Monthly Batch	These scripts are executed less frequently like once-a-week or once-a-month.

Note: The system currently does not permit Intra-day batch runs without overnight batch run. There must be an overnight/regular batch run first before the Intra-day batch runs can be scheduled. If no Intra-day batch run is desired, the overnight/regular batch scripts should be executed in regular mode only. Table 6–2 lists all the important batch scripts in sequence along with their execution frequency.

Execution Frequency of AIP Java/Oracle Batch Scripts

Table 6–2 provides information about the Execution Frequency of the AIP Java/Oracle batch scripts.

Table 6–2 AIP Java/Oracle Batch Scripts

Script Call	Description of Action	Initial Batch Run	Execution Frequency
batch_lock.sh Examples: batch_lock.sh -l 7 (release wave 7 lock) batch_lock.sh -l -1 (overnight lock)	Sets a lock indicating that batch is running. The lock prevent users from performing any save operations during overnight batch or from releasing orders early that are being re-planned Intra-day.	No	Intra-day, Daily
scripts/vdate.sh Examples: vdate.sh, vdate.sh inc, vdate.sh set export 20110331	Set/Get a specific AIP batch date. When used with inc, increment existing date. VDATE remains same during AIP-Oracle and AIP-RPAS batch processes even when actual systems date changes.	Yes	Daily
cron_export.sh	Exports all the DM and hierarchy information from AIP-Oracle to AIP-RPAS. Prepares Oracle database for next scaling and smoothing runs.	No	Daily
cron_import.sh dm	Imports all the DM and hierarchy information from AIP-RPAS into the AIP-Oracle database.	Yes	Daily
cron_import.sh sku-attributes	Imports the SKU cost, weight, volume information from external system into AIP-Oracle for PO scaling.	Yes	Daily
cron_import.sh om	Imports OM from external system into AIP-Oracle data for updating order details in the Oracle database.	No	Daily
pre_scale.sh	Prepares database for scaling the warehouse POs that will be loaded from AIP RPAS. Required only when scaling is desired.	No	Daily
prepare_for_intra_day_release.sh -w <wavenumber>	This script prepares Oracle database for loading and releasing given Intra-day wave. It should be run every time before loading Intra-day wave orders. Not needed for overnight load and release.	No	Intra-day
cron_import_order.sh -o transfer -d store -w <wavenumber>	Imports RPAS generated into-store transfers into AIP Oracle tables. When optional parameter wavenumber is passed, the script runs in Intra-day mode.	No	Intra-day, Daily
cron_import_order.sh -o transfer -d warehouse	Imports RPAS generated overnight into-warehouse transfers into AIP Oracle tables.	No	Daily
cron_release.sh -o transfer -d store -w <wavenumber>	Release into-store overnight or Intra-day (when optional parameter wavenumber) transfers to the tsf_mfqueue table.	No	Intra-day, Daily
cron_release.sh -o transfer -d warehouse	Release into-warehouse overnight transfers to the tsf_mfqueue table.	No	Daily
cron_import_order.sh -o purchase -d store -w <wavenumber>	Imports RPAS generated into-store overnight or Intra-day (when optional parameter wavenumber) purchase orders into AIP Oracle tables.	No	Intra-day, Daily
cron_release.sh -o purchase -d store -w <wavenumber>	Release into-store overnight and Intra-day purchase orders to the po_mfqueue table. When optional wavenumber parameter is passed, the script runs in Intra-day mode releasing Intra-day orders.	No	Daily, Intra-day

Table 6–2 (Cont.) AIP Java/Oracle Batch Scripts

Script Call	Description of Action	Initial Batch Run	Execution Frequency
cron_import_order.sh -o purchase -d warehouse	Imports RPAS generated into-warehouse overnight purchase orders into AIP Oracle tables.	No	Daily
smooth_order.sh	Performs Order Smoothing after importing overnight into-warehouse POs.	No	Daily
scale_order.sh	Performs Supplier Scaling and/or Container Scaling after smoothing.	No	Daily
merge_order.sh	Moves scaled into-warehouse POs to the final database table for Release.	No	Daily
cron_release.sh -o purchase -d warehouse	Release into-warehouse overnight purchase orders to the po_mfqueue table.	No	Daily
tsf_po_export.sh -o all -d all	Execute only when overnight released orders are published to RMS through a flat file.	No	Daily
post_scale.sh	Exports modified warehouse POs as a result of smoothing and scaling and executed SPQ quantities to AIP-RPAS.	No	Daily
cron_post_release.sh -o all -d all	Performs following overnight post-release activities: Identifies overdue orders, Updates order history of into-warehouse orders, Refreshes into-warehouse forecast transfers in non_ contents_order table.	No	Daily
batch_lock.sh -u	Clears the overnight or Intra-day lock set earlier.	No	Intra-day, Daily
cron_purge.sh -o all -d all	Deletes closed purchase orders and/or transfers that have exceeded their defined purge age. This script can be run in non-critical window.	No	Daily
cron_import.sh alerts	Imports DM alerts generated by AIP RPAS batch.	Yes	Daily
purge_log.sh	Purges the PLSQL batch log data. This can be run in non-critical window.	No	Daily
scripts/aip_purge.sh	Depending upon purge age parameters, this script purges the invalid or ineffective hierarchy and measure data from Oracle tables in order to maintain database tablespace. This can be scheduled to run less frequently like once a week or once a month.	No	Weekly or Monthly
generate_sched.sh	Generates order schedule based on order cycles and supply chain constraints.	No	Daily
cron_export_sched.sh	Extracts schedules for AIP RPAS.	No	Daily
cron_export_dcrp.sh	Extracts detailed cross dock constraint receipt plan AIP RPAS use.	No	Daily, Intra-day

Directory Structure

In addition to the scripts listed in Table 6–2, there are some additional control scripts and directories present in the integration home of AIP Oracle platform. Table 6–3 lists the extra files and directories in integration home.

Table 6–3 Directory Structure for the AIP Java/ Oracle Platform

Files and Directories	Explanation
aip_common_online.sh	This is a common script sourced by all AIP-Oracle shell scripts.
aip_env_online.sh	This is environment script that has DB alias, location of RETL configuration file, and so on. This script is sourced by aip_common_online.sh.
config.xml	The RETL configurations file containing database connection information.
data/	The directory that is used to store the tarred and compressed inbound data files for loading into AIP online by cron_import.sh and cron_import_order.sh scripts.
schema/	The directory of schema (xml) files used by RETL to load into and extract data from AIP online tables.
scripts/	The location of automation scripts for data processing.
temp/	The temporary folder required for Batch Script Architecture (BSA).
config/	The location of configuration files, such as import_dm.config, and so on.
logs/	The location of log files.
archive/	The location of the archived input data files.

Prerequisites for AIP Java/ Oracle Platform

The machine running interface scripts must have the ability to schedule jobs to execute scripts on a timed schedule.

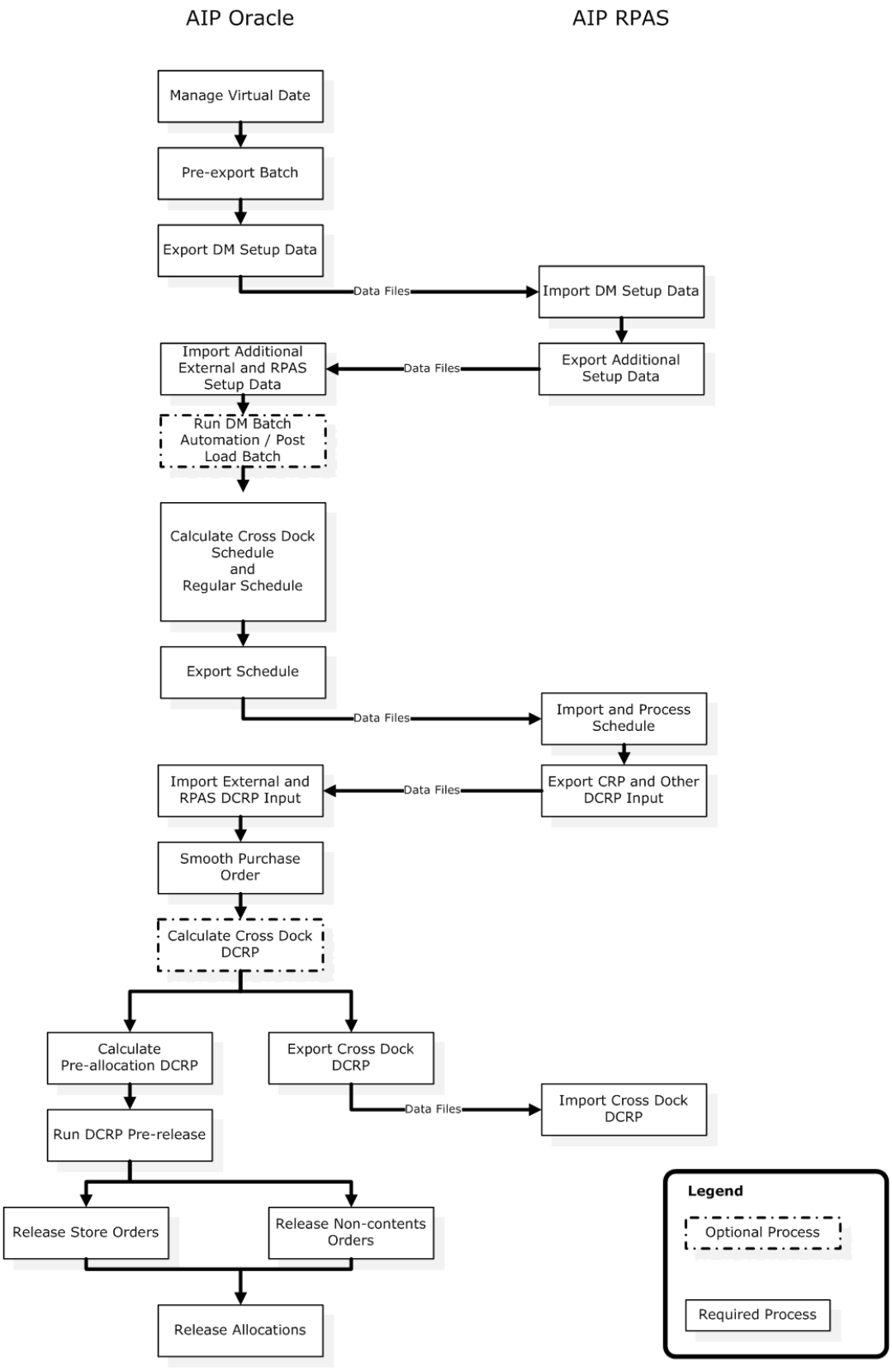
RETL must be installed and available in the PATH. See the section, "Compatibility and Hardware Requirements" of the *Oracle Retail Advanced Inventory Planning Installation Guide* for the supported version of RETL for this release.

The database schema must be installed and built with all interface procedures compiled and available. The database credentials should be secured in Oracle Wallet. Various parts of AIP-Oracle application assumes the availability of wallets alias to access the database.

AIP Java/Oracle Batch Process Flow

Figure 6–1 illustrates the AIP Java/Oracle batch process flow.

Figure 6–1 AIP Java/Oracle Batch Process Flow



Batch Process Description

The Oracle batch process consists of a series of data processing and transport (import/export) to and from AIP RPAS.

It begins with AIP Oracle sending Data Management setup information to AIP RPAS. AIP RPAS in turn uses that information to generate other setup information and exports them back to AIP Oracle to be used for schedule calculation.

Once AIP Oracle generates schedules, they are sent back to AIP RPAS. AIP RPAS uses schedule date to generate constrained receipt plans (CRP). CRPs are passed back to AIP Oracle to generate detailed constraint receipt plans (DCRP). DCRPs are then used as is input to order release and also sent back to AIP RPAS for further processing.

Into-store orders and into-store transfers set for replan can be replanned and released during the day (Intra-day) after overnight orders are released. While Intra-day replanning happens on the AIP-RPAS side, the Intra-day release occurs on the AIP-Oracle side. During Intra-day batch, only the replanned orders generated in AIP-RPAS are exchanged between the two sides.

Export from AIP Online to RPAS

To export from AIP Online to RPAS, perform the following procedure:

1. At the end of the business day, user-entered data from the AIP Online application must be transferred to RPAS before AIP RPAS batch begins.
2. The cron_export.sh process begins by running a pre-export, pre_aiponline_ extract.sh, script. This script runs clean-up in the outbound directory for any data files from a previously failed export run. If cron_export.sh completes successfully, there should be no data files left in the outbound directory. The presence of these files can end up creating incorrect *.tar.Z files.
3. Step 2 is followed by a SQL query that resets required parameters in the Oracle database, including date and horizon information.
4. Data is then extracted from the appropriate tables and stored in .dat files. When the export is complete, the .dat files are compressed and tarred into .tar.Z files. and then waits for retrieval to be processed by RPAS interface scripts. In between the hierarchy and DM export calls, PLSQL functions are called to build scalable and smoothable assignments to prepare for next smoothing and scaling batch run.
5. After export is complete, the *last_export* parameter is reset in the Oracle database.
6. A log file is generated at ${INTEGRATION_HOME}/logs.

Import from RPAS and External System into AIP Online

To import from RPAS and external system into AIP Online, perform the following procedure:

1. After AIP RPAS batch has successfully completed, the AIP RPAS interface processes generate files for loading into the Oracle-based AIP Online application.

   By default, these files are compressed and placed in the AIP RPAS domain. Specifically, the files are placed in the $AIPDOMAIN/interface/export directory. The names of the files are `dm.tar.Z`, `dm_alerts.tar.Z`, and `srp.tar.Z`.

   Both overnight and Intra-day orders use the same tar filename, that is `srp.tar.Z`. All of these files are created and processed at different times of batch processing.

2. The inbound files from the RPAS server should be retrieved by FTP using a job scheduling application at different times.
3. Once files are successfully transferred and the scheduled job begins the cron_import.sh or cron_import_order.sh scripts, the files are copied to $BSA_ARCHIVE_DIR with new file names carrying timestamp. For example: `dm.tar.Z.<timestamp>`, `dm_alerts.tar.Z.<timestamp>`, and `srp.tar.Z.<timestamp>` and so on.
4. These files are unpacked into $ONL_INBOUND_DIR and the original tar.Z archives are removed.
5. Once the data files are extracted from the loading scripts cron_import.sh or cron_import_order.sh load the data from .dat files into the Oracle tables.

   Cron_import.sh loads the `dm.tar.Z` and `dm_alerts.tar.Z`. cron_import_order.sh loads the `srp.tar.Z`.
6. A log file is generated at ${INTEGRATION_HOME}/logs.
7. External systems also provide data that loads directly into the Oracle database. These files (such as recycled order number, intransit/received quantities, sku-attributes, and so on) must be placed in the ONL_INBOUND_DIR by a job scheduling application task or the sending application.

   These data files are loaded into Oracle database using cron_import.sh script.

Smooth and Scale Purchase Orders

After AIP-RPAS generated overnight warehouse purchase orders are loaded into Oracle tables, that is after completing cron_import_order.sh for overnight warehouse purchase orders, they go through *order smoothing* module followed by *order scaling and container scaling* module. Both order smoothing and scaling modules attempts to move orders to earlier delivery dates in order to meet their respective constraints.

Before smoothing occurs pre-scaling process prepares the database for scaling and smoothing prior to entering the critical batch window. At the completion of scaling the orders which have been moved or modified by smoothing or scaling are exported out of the oracle database and archived in the ocs.tar.Z file. This file must be moved to the AIP RPAS inbound directory for the final load step.

Purchase Order/Transfer Release from AIP Online to Merchandising System

The into-store and into-warehouse PO/TSF release processes are both preceded by validation to ensure that order definitions exist, order purge periods exist, and so forth. Both processes populate the PO_MFQUEUE and TSF_MFQUEUE tables, which are queried by the OrderSenderBean.

The Intra-day and overnight batch follow the same data flow in this regards.

7

AIP Java/Oracle Daily Batch Process Details

The batch scripts are executed on the AIP Oracle database before and after the AIP batch execution on the AIP RPAS platform.

The batch scripts initiate a number of processes serially. It is accepted that they will not provide optimum parallelization and restart-ability/recovery. The restart/recovery of the batch from the highest level scripts requires a significant amount of the batch to be rerun. You have the option of manually running the lower level scripts from the failing script onward, however it is not recommended in production. For this reason, the second level scripts are modularized such that they can be executed by the job scheduler, with the appropriate dependencies, and run individually (followed by the scheduled dependent processes).

This chapter describes the following sections:

Locking and Unlocking Users

Online users should be locked out from making changes to data that is going to be processed by the batch processes. In order to do this, a script called batch_lock.sh is provided in the home directory of AIP Oracle batch scripts. This script is used with different parameters for both locking and unlocking users.

Client's job scheduler should execute the script to lock users every time before the start of overnight or Intra-day batch processes. When locking for Intra-day batch, only the release wave that is going to be processed should be locked. And then execute the script again to unlock at the end of batch processes so that users can resume online activities. While overnight batch lock is more restrictive than Intra-day batch lock, the latter applies only to the data that will be processed in the locked release wave.

The overnight batch processes on the AIP-Oracle usually begin with the process that increments VDATE before exporting data from Oracle to AIP-RPAS and ends at the post release process. The Intra-day batch processes usually begin with the process that

captures inventory snapshot in external systems such as RMS and ends at the release of Intra-day orders.

Script Name: batch_lock.sh

This script calls a PLSQL package function aip_util.set_bach_lock to set or clear the lock. Inside the function, system parameters BATCH_LOCK and BATCH_LOCK_RELEASE_WAVE are updated with values depending upon the input parameters passed to the script. Valid values for BATCH_LOCK are Y and N, Y indicates the lock is in place, N means no lock. BATCH_LOCK_RELEASE_WAVE is -1 for overnight and a number between 0 and 23 for Intra-day release wave.

Input Parameters:

Following are the input parameters for the batch_lock.sh script.

- Lock option (l): value -1 for overnight lock, a value between 0 and 23 for Intra-day release wave.
- Unlock option (u): no value. Clears all locks.

Examples:

- batch_lock.sh -l -1: Overnight lock
- batch_lock.sh -l 7: Intra-day release wave 7 lock
- batch_lock.sh -u: Unlock

Virtual Date Maintenance

The entire batch process is run for the next business day. To guarantee that the whole batch process uses the same date, and is not affected by a potential change in calendar days on the server, a virtual date is used.

The virtual date, also referred to as the batch run date or vdate, is set immediately prior to the commencement of the batch. If the scheduled start time is prior to midnight the virtual date is one day ahead of the system date. If the scheduled start time is after midnight the virtual date matches the system date. On normal, day-to-day batch runs the virtual date can simply be incremented by one day to achieve the proper batch run date (regardless of scheduled start time).

A script is used to maintain the vdate. The maintenance of the vdate can be easily accomplished as a scheduled daily task and as a manual task.

The system operator schedules the increment of the vdate on a daily basis.

In the event of a batch failure all or portions of the batch may be re-run without changing the virtual date. No other batch process attempts to update or maintain the vdate. The scheduler and system operator are solely responsible for this maintenance.

The system operator is also able to manually set the vdate. Any attempt to set an invalid date results in an error being logged; the date is not updated. If the operator updates the date directly rather than through the vdate.sh script provided any attempt to retrieve or use the invalid date results in an error that halts the calling process and all dependent processes.

All batch processing dependent on dates (effective dates or end dates) considers the vdate as the current effective date. The export planning horizon and any other calculated time periods (that is, sister store/warehouse copy, walking order cycles, and so forth.) are calculated based on the vdate as the current effective date.

Script Name: vdate.sh

This script is a wrapper to the vdate functions, the get_vdate, set_vdate and inc_vdate from the package aip_util.

> **Note:** The vdate.sh script should not be used to set or increment the vdate more than once in a single, complete batch run.

Input Parameters

This script accepts the following commands:

- Action: [set | get | inc]
- Export: [noexport | export]
- Date: [YYYYMMDD] (only accepted for the set action)

set [date in YYYYMMDD]

This function performs the following:

- Calls the SET_VDATE function (using call_bsa_sql.sh).
- If export= true, write vdate value to text file.

get

This function calls the GET_VDATE function (using call_bsa_sql.sh).

inc

This function performs the following:

- Calls the INC_VDATE function using call_bsa_sql.sh).
- If export=true, write vdate value to text file using GET_VDATE function output.

Example

Script Call	Explanation
vdate.sh	retrieves the vdate
vdate.sh get	retrieves the vdate
vdate inc	increments the vdate, without transferring it
vdate inc noexport	increments the vdate, without exporting it
vdate inc export	increments the vdate and exports it
vdate set noexport 20001130	sets the vdate, without exporting it
vdate set export 20001130	set the vdate and exports it

Pre-Export Batch

Prior to physically exporting the DM Oracle data out of the tables, the calculation of the walking order cycles lead time is performed along with the export planning horizon start and end date update.

Export Planning Horizon

The export planning horizon is a calculated planning horizon value that is used to limit the amount of data exported to AIP RPAS. The export planning horizon start date is equal to the virtual batch run date (vdate). The export planning horizon end date is equal to the start date plus the calculated maximum export planning horizon value. These values are used throughout the extract logic to limit the extracted data to that which falls within the export planning horizon start and end dates. In addition to calculating maximum planning horizon, individual SKU planning horizon is also calculated in this step. The horizon itself is exported to AIP-RPAS for use in AIP-RPAS batch processes.

It is important to note that this export planning horizon does not control replenishment. However the data that is exported directly effects the replenishment plan batch calculates. Therefore, it is pertinent that all data is exported for the duration of its corresponding planning horizon.

Walking Order Cycles

Walking order cycles are used to drive stock into a new store prior to its opening. A profile/store release schedule exception is created with a special empty order cycle to wipe out all lead times for the store. Then, a second calculated release schedule lead time exception is created. This is the *walking order cycle lead time* which is recalculated each day.

Because the user has the opportunity to create new Profiles during the online day the profile/store release schedule exception must be created for the new profiles after the online day. The profile/store release schedule exception is created for all new profile/store combinations prior to the export of DM Online data to AIP RPAS.

Prior to exporting the DM Online data for running the AIP RPAS batch calculations the *walking order cycle lead time* is calculated for the current batch run date (the next business day). This lead time is saved as a SKU/store release schedule exception.

Build Scalable and Smoothable Assignments

If smoothing functionality is turned ON then smoothable assignments are identified and set up in preparation for future smoothing run. Similarly, if scaling functionality is turned ON then scalable assignments are identified and set up in preparation for future scaling run. One of two global scaling flags needs to be turned ON. Active scaling assignment are considered scalable depending upon the local (scaling group level) flags, scaling horizons and scaling constraints.

Smoothable assignments from last export run are deleted and new smoothable assignments for upcoming smoothing run are set in the smoothable_assignment table. Similarly scalable assignments from the last export run are deleted and new scalable assignments for upcoming scaling run are set in the scalable_assignment table.

During export, a list of distinct vendor/SKU from scalable and smoothable assignments is exported to AIP-RPAS so that AIP-RPAS can return warehouse release schedule and other required information only for the vendor-SKU listed in the file.

Script Call

These steps are performed just prior to the physical export of data but are executed by the same control script which triggers the exports. The control script is described in detail in the next section.

Export DM and OM Data (cron_export)

The DM Online and Order Management applications are used to enter and maintain AIP supply chain configurations, replenishment parameters, and purchase orders. While all of the user entered data is mastered in the Oracle database, it must ultimately be used to generate the desired replenishment plans.

Technical Details

cron_export is the wrapper script executed to export all the data from DM for the AIP RPAS platform. It performs the following activities:

1. Calls the pre_aiponline_extract.sh script to do the following activities:
   - Remove all the .dat and .int files in the ONL_OUTBOUND_DIR directory.
   - Call the pre_extract_wrapper function from the aip_util package.
2. Processing and archiving hierarchy and DM data.

   The script loops through the following configuration files and executes each export script listed in the file:
   - $INTEGRATION_HOME\config\export_hierarchy.config
   - $INTEGRATION_HOME\config\export_dm.config
3. Build smoothing and scaling data.

   The script calls build_scalable_assignment and build_smoothing_data functions in ocs_validation package to prepare assignments for next batch run of scaling/smoothing.
4. Updating Interface Parameters.

   The update_interface_param.sh script updates the last used planning date for SKU planning horizon in planning_horizon_sku table and MAX planning horizon in interface_parameters table.

Files to Export Data

The following files are needed to export data out of the AIP Oracle schema.

File	Description
Export Script ($INTEGRATION_HOME/scripts/export)	This folder contains the export SQL query.
Schema File ($INTEGRATION_HOME/schema)	Defines the .dat file format using xml tags.
Config File ($INTEGRATION_HOME/config)	Lists the .sh export scripts that can be executed in parallel.

Tables to Extract Data

Two key tables are used to extract data from the Oracle tables.

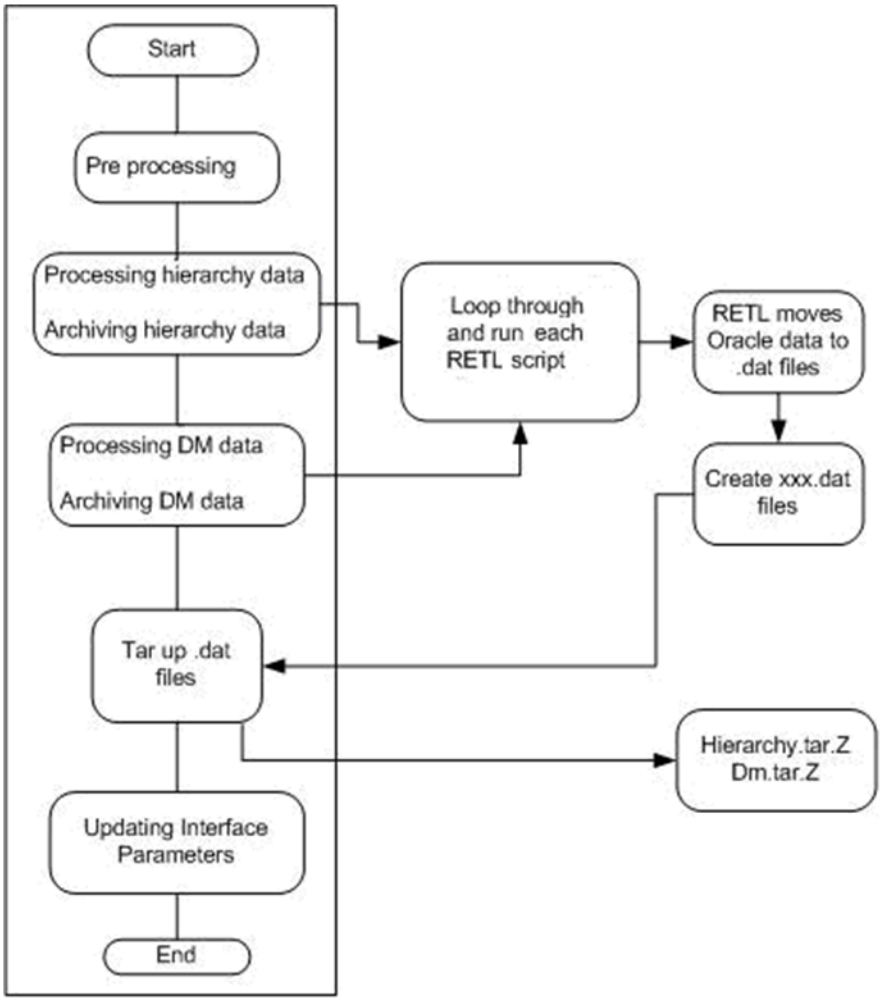
Table	Description
INTERFACE_PARAMETER	The INTERFACE_PARAMETER table holds the timestamp of the last successful run of the cron_export.sh. This is how RETL knows what was added or modified since the last extract. In many cases, only modified data is extracted.

Table	Description
INTERFACE_HORIZON_DATE_ LOOKUP	The INTERFACE_HORIZON_DATE_LOOKUP table has one row for each day starting today through the entire planning horizon (as calculated in the pre-export logic). This table is used as a join in the SQL statement within the export script. It is used to blow out the data to the daily level or to ensure that the data being exported falls within the planning horizon.

Export Flow Diagram

Table 7–1 shows the flow of data out of the AIP Oracle schema.

Figure 7–1 Export Flow



Script Name: cron_export.sh

This script exports all the DM and hierarchy information from AIP-Oracle to AIP-RPAS. This script prepares Oracle database for next scaling and smoothing runs.

Input Parameters

noTimestamp is an optional parameter that indicates whether to update the INTERFACE_PARAMETER table or not. By default, the table is updated unless the value noTimestamp is entered to prevent the LAST_EXPORT value from being updated.

Requirements

This script requires a configuration file containing a list of export scripts.

Day on Day Processing

The script exports data from the Oracle database to AIP RPAS. The script is designed to execute automatically by a scheduled CRON job every night after the online day and before AIP RPAS Batch is run.

> **Note:** It is assumed that batch lock has been applied and VDATE has been updated as documented earlier before the start of this process.

The script first calls pre_aiponline_extract.sh to perform pre-export updates. Next, each export script is invoked. After the data is successfully extracted, the script updates the timestamp on the INTERFACE_PARAMETERS table.

Processing Steps

Table 7–1 lists the processing steps for cron_export.sh. Details of these steps are described in their corresponding section.

Table 7–1 Processing Steps for cron_export.sh

Step	Description
1	Execute pre-extract setup and maintenance.
2	Process hierarchy data.
3	Archive hierarchy output files.
4	Build scalable and smoothable assignments.
5	Process DM data.
6	Archive DM output files.
7	Update the interface parameters.

1. Execute pre-extract setup and maintenance.

   The pre_aiponline_extract.sh script sets up the planning horizon days which are used throughout the extract logic and maintains/decrements the walking order cycle exceptions.

2. Process hierarchy data.

   Use the process_aiponline_data.sh script with the input parameter as export_hierarchy.config to process the hierarchy data.

   Input Parameters

   Following are the input parameters for the process_aiponline_data.sh script.

   - export_hierarchy.config
   - ${ONL_EXPORT_DIR}

Details of Processing Hierarchy Data

Each export script within the configuration file makes a single call to RETL with a database connection parameter. The export scripts contain the export SQL query. When more than one file is being generated, the export script contains one query for each unique file name being generated.

Hierarchy data included in the export_hierarchy.config file are:

- network_groups
- order_cycle
- order_groups
- profile
- sku_pack
- store_order_cycle
- warehouse

3. Archive hierarchy output files.

   After processing the hierarchy data using the previous step, the data is archived by converting the hierarchy (.dat) files into tar files.

   For example, hierarchy.tar.Z file.

4. Build scalable and smoothable assignments.

   Use PL/SQL stored procedure to build a new set of scalable assignments if scaling is turned ON. Also call stored procedures that builds smoothing data like smoothable assignments, smoothing detail when smoothing is turned ON.

5. Process DM data.

   Use the process_aiponline_data.sh script with the input parameter as export_dm.config to process the DM data.

   Input Parameters

   Following are the input parameters for the process_aiponline_data.sh script.

   - export_dm.config
   - ${ONL_EXPORT_DIR}

   Details of Processing DM Data

   The following table lists the DM data included in the export_dm.config file.

DM Data Included in the export_dm.config Files:		
assigned_commodity	non_release_date	sister_store
chamber_product_type	non_release_date_exceptions	sister_wh
commodity_order_cycle_exceptions	order_cycle	stockless_indicator
default_order_cycle	order_group_assignment	store_calendar
delivery_demand_percent (4 levels)	order_multiple	store_format_pack_size
delivery_pattern	orderable_unit	store_order_cycle
assigned_commodity	orderable_unit	singles_enabled_warehouse
chamber_product_type	pack_break_size	sister_store

DM Data Included in the export_dm.config Files:		
delivery_demand_percent_location	pack_break_warehouse	sister_warehouse
delivery_demand_percent_location_department	pack_change	source_split_tb
delivery_demand_percent_location_department_exception	pack_change_clear_inventory_period	stockless_indicator
delivery_demand_percent_location_exception	pack_change_destination	store_format_pack_size
demand_group_data	pallet_multiple	store_planning_horizon
department_profile	profile_network_group	store_priority
direct_store_format_pack_size	profile_order_cycle	supplier_order_cycle_assignment
direct_store_pack_size	promotion_and_replace	valid_warehouse
direct_supplier	promotion_and_replace (standard SKU,store and source)	vendor_lock
inventory_snapshot_time	range_commodity_pack_size	vendor_sku_to_scale
on_supply_off_supply	rdc_reconciliation	warehouse_prealloc_mustconsume (global,class and sku)
order_group_assignment	release_wave_assignment (default and exception)	warehouse_prealloc_preallocate_sku
order_history	replan_flag_default	warehouse_prealloc_stagingwindow (global,class and sku)
order_multiple	secondary_source	warehouse_rollout_status

6. Archive DM output files.

   After processing the DM data using the previous step, the data is archived by converting the DM (.dat) files into tar files. For example, dm.tar.Z file.

7. Update the interface parameters.

   The update_interface_param.sh script updates the INTERFACE_PARAMETERS table to reflect what the timestamp was at the end of this export.

   Due to the fact that some data is extracted, or maintained at high volumes, or both, that data is only exported when changes occur. In order to identify changes that occur since the last export, a last export timestamp is maintained. This timestamp is updated immediately after all export data files have been successfully created and compressed.

   All subsequent updates made in the DM Online application, or as part of the AIP Oracle Morning Batch, have a later timestamp and are therefore picked up in the next export.

   Details of Updating Interface Parameters

   When no parameter is passed into cron_export.sh, the last_export timestamp is updated to the current date and time. When the noTimestamp parameter is passed to cron_export.sh, the last_export timestamp is not updated.

Prerequisites for running cron_export.sh

Run this step after the DM and OM Online users are locked by batch_lock.sh -l -1 execution. This prevents any opportunity for you to add or modify data between the

time that data is extracted from a table and the last export timestamp is updated. Such an occurrence could result in AIP Oracle and AIP RPAS getting out of sync.

The data extracted also depends on the INTERFACE_HORIZON_DATE _LOOKUP table and the INTERFACE_PARAMETER table. If data is not extracted as expected, ensure that the days_in_export_horizon value in the Config package is set correctly to cover the longest planning horizon in the AIP system.

The data files that do not contain a full refresh of the data are controlled by the last_ export timestamp value. The timestamp on the row being extracted is compared to the last_export timestamp. If the timestamp on the row is later than the last_export value, it was added or modified since the last extract occurred.

This step must run before the AIP RPAS batch step prep_onl_data.

Restart/Recovery steps for cron_export.sh

If this step fails, perform the following:

1. Review the log file, which is located in the logs folder of the directory that is specified for the Oracle LOGHOME environment variable.
2. Perform the necessary corrective actions.
3. Re-run cron_export.sh.

Import Data into AIP Oracle Database

The following sections describe how to import data into the AIP Oracle Database.

Load Data

Much of the AIP supply chain management is performed in Data Management Online, which is built on an Oracle database. The RMS and RPAS data is imported into DM Oracle as part of the final steps of the AIP nightly batch.

The import includes foundation data and the orders that were produced by AIP RPAS batch. Once imported, today's orders are scaled and immediately released to RMS by OM batch. The release to RMS serves to execute the orders or to communicate them to the source warehouses and suppliers.

Import Overview

Upon completion of the batch jobs, RPAS generates flat files (.dat data files) that contain the hierarchy and measure data that is required in DM and OM Oracle Online. Data files exported from RPAS should then be retrieved by copy or FTP using a job scheduling application. Once the Oracle import process begins, the data is loaded into the Oracle database by using RETL (Oracle Retail Extract, Transformation, and Load). For this step, only the Load part of RETL is used.

Figure 7–2 Importing Flow

Staging tables, prefaced with **INTERFACE_** or **i_**, temporarily store the records from the .dat file in the Oracle database. RETL then invokes a stored procedure to move the data from the staging table to the base table.

The following two files are needed to move the flat file to the staging table using RETL:

- Import Script ($INTEGRATION_HOME/scripts/import)
- Schema File ($INTEGRATION_HOME/schema)

Technical Details

The cron_import.sh script currently performs two main functions:

- Import Hierarchy, DM, SKU attributes, DM Alerts, and OM data into the AIP Online database (process_aiponline_data.sh).
- Call any post-load processing if necessary (post_import_wrapper.sh).

This script is a wrapper for a series of scripts.

The importing of data into AIP Online can actually run as independent processes, as listed in the following table.

Process	Steps
Import Data Management files	■ Retrieves the file containing DM data ■ Calls the import processing script with the hierarchy config file ■ Calls the import processing script with the dm config file ■ Calls the post-load SQL processing script to execute the Automated Data Maintenance batch scripts
Import SKU Attribute data	■ Retrieves the file container SKU Attributes from the import directory ■ Calls the import processing script with the sku-attributes config file
Import Order Management data	■ Retrieves the file containing OM data ■ Calls the import processing script with om config file
Import Data Management Alerts	■ Retrieve the file containing DM Alerts data ■ Calls the import processing script with dm alerts config file

Script Name: cron_import.sh

The script, cron_import.sh is executed to process the files by passing the appropriate parameters.

Input Parameters

Following are the input parameters for the cron_import.sh script.

Module name: [dm | alerts | om | sku-attributes]

Restart/Recovery

If this script fails, perform the following steps:

The import process executes many import scripts in parallel. If this fails, it is because one or more of the import scripts failed. After examining the logs to determine which scripts failed, and after correcting the errors, this process can be completed in a number of ways:

- Simply restart the import processing. Since successful import scripts deletes their input files, the entire import process can be run again. Missing input files only generate warnings and then continue.
- Execute individual import scripts from the command line.
- Create a config file containing the list of only the import scripts that have not completed successfully. Execute the import process using this config file.

If an error occurs in the post-load SQL processing script, then once the errors have been corrected that script can simply be re-executed.

DM Post-Load Batch

The following sections describe the process of DM Post-Load Batch.

Script: post_import_wrapper.sh

The DM scripts that are executed by cron_import.sh after the AIP Oracle load scripts are called DM Post-Load batch. It is also referred to as Automated Data Maintenance Batch.

The following descriptions of each process provide an overview. Additional validation and restrictions may be applied that are not detailed in the overview.

Create Order Groups

The automatic creation of order groups is triggered by a new supplier arriving in the AIP system. When the new supplier has a ship-to value defined it is compared against a mapping table. The mapping table maps ship-to values to sources/source warehouse types and destination warehouse types. One order group is created for each source and destination warehouse-chamber combination assigned to the ship-to source type and destination warehouse-type. The chambers assigned to warehouses that match the ship-to destination warehouse types are assigned as the order group automation scheduling locations.

If the new supplier supplies SKU-packs that already exist in the system (SKU-packs are multi-supplied by another existing supplier), the demand groups of the SKU-packs are immediately assigned to the order group.

Create Profiles

The automatic creation of profiles is triggered by a new supplier arriving in the AIP system. The new supplier triggers the creation of a new direct profile by DM Oracle batch. The new direct profile is assigned a specific default order cycle which is created at implementation time.

Note: The automatic creation of warehouse profiles occurs in the AIP RPAS batch.

Assign Profiles

The assignment of SKUs to profiles must be performed after the automatic creation of profiles. The automatic assignment of SKUs to profiles is triggered by a new Supplier/SKU-pack size combination. The SKUs which are identified as newly supplied by a supplier are assigned to that supplier's direct profile. If the supplier does not have a direct profile, and the SKU is not assigned to any profiles, the SKU is assigned to the Default (Class) Profile where defined.

Note: The automatic assignment of SKUs to warehouse profiles is performed in AIP RPAS batch and is triggered by a new SKU.

Demand Group and SKU Group Maintenance

The following sections describe demand group and SKU group maintenance.

New SKU-pack Sizes

All new SKU-pack sizes that arrive in the AIP system are assigned to a single default demand group and SKU group. These SKU-pack sizes cannot be replenished in the warehouse until they are assigned to the proper demand group and SKU group.

A new SKU group is created for each new SKU.

The assignment of new SKU-packsizes to demand groups is determined by a configuration option. This option, set at implementation time, determines if the new packsizes of a SKU should be in a single demand group for that SKU, or each packsize should have its own demand group.

The default configuration is to assign all of a SKU's pack sizes to the same demand group.

An additional parameter—the Inventory Tracking Level— can override the demand group assignment configuration option. The Inventory Tracking Level is a global string value set at implementation time. When it is set to Eaches the demand group assignment configuration option previously mentioned is over-ruled, and all new packsizes of a SKU must be assigned to a single demand group for the SKU.

Pre-priced SKU-pack Sizes

All SKU-pack sizes that are pre-priced/value-added have a parent SKU defined. The pre-priced/value-added SKUs must be assigned to the same SKU group and demand group as their parent SKU.

Any pre-priced/value-added SKU which is assigned to a SKU group other than its parent's SKU group is updated to reflect a SKU group assignment equal to the parent's SKU group.

Any pre-priced/value-added SKU-pack size which is assigned to a demand group other than its parent's demand group is updated to reflect a demand group assignment equal to the parent's demand group. Because a SKU's pack sizes may be assigned to multiple demand groups the first available parent demand group (when ordered by the demand group code) is used for all of the pre-priced/value-added SKU-pack size assignments.

Standard SKUs

All SKUs that have changed status from pre-priced/value-added to a standard SKU must be assigned to their own demand group and SKU group.

A new SKU group is created for each SKU which is now a standard SKU.

A new demand group is created for each new standard SKU or SKU-pack size (depending on the configuration setting)

Clean-up

When SKU-packs become pre-priced/value-added, they are moved into their parent's demand group and SKU group. Moving SKU-packs out of an existing demand group

may result in an empty demand group. All demand groups which have no SKU-packs assigned to them, with the exception of the default demand group, are deleted.

Range SKU-pack sizes

Automatic ranging is triggered by a new SKU-pack size. Any SKU-pack size that was not in the AIP system in the previous batch run is considered new. The new SKU-packs are ranged to warehouses based on a configurable system parameter setting that allows ranging based on the SKU's supplier's ship-to value and order group destinations or all valid warehouses. The ranging rules applied in the DM Online application are also applied to the batch process.

Order Group Assignment

Automatic order group assignment is triggered by a new SKU-pack size. The assignment of demand groups to order groups must be performed after the automatic creation of order groups, after SKU group and demand group maintenance, and after automatic ranging.

New SKU-pack sizes' demand groups are assigned to order groups through their supplier's ship-to value. All order groups with sources that match the supplier's ship-to source value are retrieved. The new SKU-pack sizes' demand groups are assigned to all order group source/scheduling location combinations which are valid based on supplier/SKU-pack size links and ranging.

Reset Store Format Pack Size (WH and Direct to Store Format)

This process should be run after Range SKU-pack sizes to ensure all available pack sizes are considered for replacing the existing pack size.

The store format ordering pack size default is reset when the current pack size has a discontinuation date equal to the vdate. The new pack size is selected in a pre-determined order based on the SKU-pack size pack type. The first valid available pack type that is not discontinued becomes the new ordering pack size default for the store format. If no alternative pack size is found the ordering pack size is not reset.

Reset Store Pack Size (WH and Direct to Store)

This process should be run after Range SKU-pack sizes to ensure all available pack sizes are considered for replacing the existing pack size.

The store ordering pack size exception is reset when the current pack size has a discontinuation date equal to the vdate. The new pack size is selected in a pre-determined order based on the SKU-pack size pack type. The first valid, available pack type that is not discontinued becomes the new ordering pack size exception for the store. If no alternative pack size is found the ordering pack size is not reset.

Set Missing Store Format Pack Size (WH and Direct to Store Format)

This process should be run after Range SKU-pack sizes to avoid a day lag in creating store format pack sizes for new SKU-pack sizes.

The default store format pack size is automatically selected for all valid source/SKU/store format combinations. The pack size is selected in a pre-determined order based on SKU-pack size pack types. The first valid, available pack type that is not discontinued becomes the ordering pack size default for the store format.

This process runs nightly to pick up all valid source/SKU/store format combinations that do not currently have a default ordering pack size defined.

Reset Warehouse Orderable Unit

This process should be run after Range SKU-pack sizes and Demand Group and SKU Group Maintenance to ensure all available pack sizes are considered for the new orderable unit.

The warehouse orderable unit is reset when the current pack size has a discontinuation date equal to or before to the vdate. The new pack size (orderable unit) is selected in a pre-determined order based on the SKU-pack size pack type. The first valid available pack type that is not discontinued becomes the new orderable pack size default for the source/demand group/SKU/warehouse. If no alternative pack size is found the orderable pack size is not reset.

Set Missing Warehouse Orderable Unit

This process should be run after Range SKU-pack sizes and Demand Group and SKU Group Maintenance to avoid a day lag in creating warehouse orderable units for new SKU-pack sizes.

The warehouse orderable unit is automatically selected for all valid source/demand group/SKU/warehouse combinations. The pack size (orderable unit) is selected in a pre-determined order based on SKU-pack size pack types. The first valid, available pack type that is not discontinued becomes the orderable pack size for the source/demand group/SKU/warehouse.

This process runs nightly to pick up all valid source/demand group/SKU/warehouse combinations that do not currently have a warehouse orderable unit defined.

Set Order Multiple

This process should be run after Range SKU-pack sizes to avoid a day lag in creating order multiples for new SKU-pack sizes.

The order multiple is automatically selected for all valid source/SKU/pack size/warehouse-chamber combinations. The order multiple is an integer value selected from the available pack sizes for the SKU. The pack size value that is used as the order multiple is selected in a pre-determined order which is based on pack types. The fist valid, available pack type that is not discontinued becomes the order multiple for the source/SKU/pack size/warehouse-chamber.

This process runs nightly to pick up all valid source/SKU/pack size/warehouse-chamber combinations that do not currently have an order multiple defined.

Set Stacking Flag

This process should be run after Range SKU-pack sizes to avoid a day lag in setting stacking flag values for new SKU-pack sizes.

The stacking flag is automatically set for all valid source/SKU/pack size/warehouse-chamber combinations. The default stacking flag value is configurable at implementation time. This process runs nightly to pick up all valid source/SKU/pack size/warehouse-chamber combinations that do not currently have a stacking flag value defined.

Note: Because this process picks up all missing values, suddenly enabling this logic on a full production set of data causes a significant increase in the batch run time

---

Set Case Weight

This process should be run after Range SKU-pack sizes to avoid a day lag in setting case weight values for new SKU-pack sizes.

The case weight is automatically set for all valid source/SKU/pack size/warehouse-chamber combinations. The default case weight value is configurable at implementation time. This process runs nightly to pick up all valid source/SKU/pack sizes/warehouse-chamber combinations that do not currently have case weight value defined.

Note: Because this process picks up all missing values, suddenly enabling this logic on a full production set of data causes a significant increase in the batch run time.

Set Pallet Multiple

This process should run after order multiples are set to prevent a day lag in setting the pallet multiple.

A pallet multiple is automatically set for all source/SKU/pack size/warehouse-chamber combinations that have an effective order multiple on the batch run date and which do not have an effective pallet multiple defined.

Create Time Balanced Source Splits

This process should run after order group assignment to prevent a day lag in creating the time-balanced source splits.

Time balanced source splits are created for every demand group/warehouse combination that has a supplier sourced order group assignment. The first supplier receives 100% of the source split percent. If more than one supplier supplies SKU-pack sizes in the demand group an alert is created to indicate that the user should review the accuracy of the source split created.

Copy Sister Store

The sister store copy logic uses a combination of the new store open date and a configurable system parameter to determine when the sister store copy occurs. When a new sister store open date is received an alert is generated to indicate that a new store open date was received. This alert also includes an expected copy date. A copy only occurs once for a particular new store. Sister store copies occur as soon as their calculated copy date is reached, and results in the supply chain of the copy to store being a replica of the copy from store.

Note: Users and custom processes must not set up the supply chain in the DM Online application for a new store with a pending sister store copy. Doing so results in a failure of the copy. The only recourse is for the batch operator to manually delete the data pertaining to the new store or the user must manually complete the entire supply-chain setup.

Because the only recourse for a failed copy is manual correction or setup it is highly recommended that the introduction of new stores with a sister store copy be a tightly controlled process. To control the process the introduction of the new stores should not occur until the store open date falls within the copy timeframe. To additionally safeguard this process the sister store offset weeks system parameter can be set to an arbitrarily large number that is less than the maximum export horizon (converted to weeks).

Copy Sister Warehouse

The sister warehouse copy logic uses a combination of the new warehouse open date and a configurable system parameter to determine when the sister warehouse copy occurs. When a new sister warehouse open date is received an alert is generated to indicate that a new warehouse open date was received. This alert also includes an expected copy date. A copy only occurs once for a particular new warehouse. Sister warehouse copies occur as soon as their calculated copy date is reached, and results in the supply chain of the copy to warehouse being a replica of the copy from warehouse.

Note: You must not set up the supply chain in the DM Online application for a warehouse with a pending sister warehouse copy. Doing so results in a failure of the copy. The only recourse is for the batch operator to manually delete the data, pertaining to the new warehouse, from the database or the user must manually complete the entire supply chain setup.

No custom processes can be created that create chambers or set ranging status for warehouses with a pending sister warehouse copy. Doing so results in a failure of the copy. All data that is created as a result of these actions must be manually deleted from the database in order for a successful copy to occur. Otherwise the user must manually complete the supply chain setup.

Because the only recourse for a failed copy is manual correction or setup it is highly recommended that the introduction of new warehouses with a sister warehouse copy be a tightly controlled process. To control the process the introduction of the new warehouse should not occur until the warehouse open date falls within the copy timeframe. To additionally safeguard this process the sister warehouse offset weeks system parameter can be set to an arbitrarily large number that is less than the maximum export horizon system parameter (converted to weeks).

Prepare for Intra-day Order Load and Release

If Intra-day replanning or Intra-day release is desired, then the Oracle database must be prepared every time before loading and releasing replanned orders for release waves. In order to save critical time, this preparation should be done while the given

release wave orders are replanned in AIP-RPAS. The preparations include identifying orders that must be removed when new set of replanned orders for the given release wave are generated by AIP-RPAS and loaded into AIP-Oracle. The data required to identify such orders is assumed to be made available to Oracle database from the overnight DM import process (cron_import.sh dm). Therefore the first execution of this script should only be done after cron_import.sh dm is completed.

A shell script, prepare_for_intra_day_release.sh, is provided in the home directory of AIP Oracle batch scripts. This must be executed once before loading orders for the given release wave. In the event that something fails on the AIP-RPAS side and new set of orders could not be made available after the preparations have been done, then the same script can be executed (with a different parameter) to undo the preparations.

Note: If Intra-day replanning or Intra-day release is not desired then this script should be skipped. Otherwise this script should be executed once for each configured release wave even when the replanning (on AIP-RPAS side) for that wave results in no new orders being generated.

Input Parameters

Following are the input parameters for the prepare_for_intra_day_release.sh script.

- Release Wave (**w**): This is a required parameter to indicate release wave for which system should prepare. Valid values are between 0 and 23.
- Undo indicator (**u**): This is an optional parameter and should be used only when something fails and the system need to undo the preparations.

Examples

prepare_for_intra_day_release.sh -w 2: This prepares for loading replanned orders for release wave 2.

prepare_for_intra_day_release.sh -w 5 -u: This will undo the preparations done for loading release wave 5 replanned orders.

Import Orders

The receipt plan that is calculated by AIP RPAS is exported from AIP RPAS upon completion of the replenishment calculations. The AIP-RPAS export configuration controls how much of the plan is exported and is therefore available to load into the Oracle database. The exported plan is eventually translated into Purchase Orders (POs) and Transfers that are sent to RMS and can be viewed and maintained. Both overnight and Intra-day replanned orders are imported into Oracle table using same cron_import_order.sh script but with different parameters.

Technical Details

The cron_import_order.sh script loads the RPAS generated orders from .dat files into AIP Online tables.

This script is parameterized so that critical orders can be loaded and released first. The non-critical orders can be loaded afterwards by passing in different parameters.

This script can be used to load the following combinations of overnight orders:

- Into-Store Purchase Orders only
- Into-Store Transfers only

- Into-Warehouse Purchase Orders only
- Into-Warehouse Transfers only
- Into-Store Purchase Orders and Into-Store Transfers (that is, all Into-Store Orders)
- Into-Warehouse Purchase Orders and Into-Warehouse Transfers (that is, ALL into-Warehouse Orders)
- Into-Store Purchase Orders and Into-Warehouse Purchase Orders. (that is, all Purchase Orders)
- Into-Store Transfers and Into-Warehouse Transfers (that is, all Transfers)
- All orders (this includes Into-Store Purchase Orders and Into-Store Transfers and Into-Warehouse Purchase Orders and Into-Warehouse Transfers)

The same script can be used to load the following combinations of Intra-day orders:

- Into-store purchase orders for a given release wave
- Into-store transfers for a given release wave
- All Into-store orders (includes into-store purchase orders and into-store transfers) for a given release wave

The into-store purchase orders are loaded from strsplrord.dat file into store_order table through staging table, interface_store_orders. When loading overnight into-store purchase orders, existing unreleased store purchase orders from previous load in store_order table are first removed. When loading Intra-day into-store purchase orders, it is assumed that required preparations for the given release wave are already done by running prepare_for_intra_day_release.sh script.

Into-store transfers are loaded from strwhord.dat file into store_order table through a staging table, interface_store_transfers in the same way as into-store purchase orders. Existing unreleased transfers are removed before overnight load and it is assumed that required release wave preparations are done before loading Intra-day orders.

Into-warehouse transfers are loaded from wh_to_wh_transfer.dat file into a staging table, i_non_contents_transfer. The cron_release.sh process moves the releaseable transfers from staging table into non_contents_order table. And then cron_post_release.sh copies the forecast transfers from staging table into non_contents_order table.

Into-warehouse purchase orders are loaded from vendor_to_wh_order.dat file into warehouse_purchase_order table through a staging table, i_non_contents_order. PO Smoothing (optional) and Scaling (optional) process update or insert new orders in warehouse_purchase_order table. Merge (required) process then copies the orders from warehouse_purchase_order to non_contents_order table.

In all of the previous cases, existing unreleased forecast orders, transfers from previous load are removed during overnight batch from main tables (store_order, non_contents_order) before new ones are added.

This script begins with fetching and extracting the order files from the compressed tar file if the tar file is available in the /data folder. Then, depending upon the order_type and dest_type parameters passed into this script, it loads the appropriate order files.

Note: Intra-day order files use the same name, that is srp.tar.Z, as the overnight order files. Therefore, care must be taken to use the correct release wave number in the loading script, cron_import_order.sh, when loading a given release wave's tar file into Oracle database.

Script Name: cron_import_order.sh

This script loads the RPAS generated orders from .dat files into AIP Online tables.

Input Parameters

Following are the input parameters for the cron_import_order.sh script.

- order_type (o): [transfer | purchase | all]. This is a required parameter for both overnight and Intra-day orders.

Parameter	Action
-o transfer	Used if transfers are to be loaded.
-o purchase	Used if purchase orders are to be loaded.
-o all	Used if both transfers and purchase orders are to be loaded.

- dest_type (d): [store | warehouse | all]. This is a required parameter.

Parameter	Action
-d store	Used if only into-store orders are to be loaded.
-d warehouse	Used if only into-warehouse orders are to be loaded.
-d all	Used if both into-store orders and into-warehouse orders are to be loaded.

- Release Wave (w): [A number between 0 and 23]. This is an optional parameter that applies to destination_type store orders only. This is used to load Intra-day orders for a given release wave.

Using a combination of order_type, dest_type and release_wave, the desired load and its priority in the critical batch window can be easily configured.

Examples of Overnight Load

The following table provides examples of overnight load.

Parameter	Action
cron_import_order.sh -o transfer -d all	Loads into-store transfers and into-warehouse transfers.
cron_import_order.sh -o purchase -d store	Loads into-store purchase orders only.
cron_import_order.sh -o all -d all	Loads into-store purchase orders, into-store transfers, into-warehouse purchase orders, and into-warehouse transfer.

Examples of Intra-day Order Loading

The following table provides examples of Intra-day order loading.

Parameter	Action
cron_import_order.sh -o purchase -d store -w 3	Loads into-store purchase orders corresponding to release wave 3.
cron_import_order.sh -o transfer -d store -w 8	Loads into-store transfers corresponding to release wave 8.

Parameter	Action
cron_import_order.sh -o all -d store -w 5	Loads into-store purchase orders and transfers corresponding to release wave 5.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch. Since successful import scripts delete their input files, the entire import process can be run again. Missing input files only generate warnings and then continue.

Smooth and Scale Purchase Orders

As soon as warehouse purchase orders have been loaded, the smoothing and/or scaling logic can be executed. These are not required by all retailers. This logic is optional depending on the business need. The execution of all or portions of this logic can also be controlled by global system parameters that disable Order Smoothing, Supplier Scaling, or Container Scaling.

Pre-Scaling

Prior to smoothing and scaling some system parameters are set that aid scaling. This script first gathers table statistics on important tables used in scaling. It then sets a system parameter to indicate whether or not any release date has multiple possible delivery dates. If quantity constraints are specified in terms of pallets, it then checks for the existence of pallet multiples and mark the invalid assignments in the list of smoothable and scalable assignments. Lastly it sets a system parameter to indicate whether or not a scaling group Supplier/SKU/warehouse has an Order Multiple that changes in the future.

Script Name: pre_scale.sh

This script prepares database for scaling the warehouse POs that will be loaded from AIP RPAS. Required only when scaling is desired.

Input Parameters

None.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the script.

Order Smoothing

The objective of smoothing order is to move orders to earlier delivery dates in order to reduce the aggregated receiving warehouse total such that it does not exceed the user defined warehouse capacity constraint. If the aggregated warehouse quantity total is already below the set limit or if no limit is defined, then no orders are moved.

Smoothing is performed on delivery dates starting from smoothing horizon in future up-to today in decreasing order of delivery dates. System Parameter GLOBAL_SMOOTHING_HORIZON is used as default global smoothing horizon in the absence of local smoothing horizon at scaling group level. Alerts are logged if smoothing is unable to reduce the warehouse quantity total to below the set capacity.

Note: For complete details on Order Smoothing, refer to the *Oracle Retail Advanced Inventory Planning Order Management User Guide*.

Technical Details

The smoothing logic first retrieves the system parameter SMOOTHING_GLOBAL_FLAG to determine whether smoothing is enabled at the global level. If smoothing is not enabled, the script completes successfully without doing anything. Otherwise, multiple threads are spawned to carry out smoothing over the SMOOTHABLE_ASSIGNMENTS table partitions. The maximum number of threads are configurable and are defined in restart_control table.

The output of smoothing is a set of modified warehouse purchase orders some of which are updated with changed quantity while some are new relative to input orders from RPAS. An intermediate table, smoothing_detail, stores the pre and post smoothed total on delivery dates that have valid capacity defined. This table is later used in scaling modules to ensure that warehouse receiving capacity is not broken in scaling also.

Script Name: smooth_order.sh

There is no input parameter to this script. When scheduling the sequence of execution, this script must be executed before scale_order.sh and after orders are loaded in warehouse purchase order table by cron_import_order.sh script. Also pre_scale.sh must be executed before executing this.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Re-execute the script. Default recovery that is built into the smoothing logic starts from the last save-point, that is, it ensures that smoothing will not attempt to smooth a successfully smoothed Scaling Group/warehouse/delivery date combination. But if for any reasons, the re-execution should redo smoothing on already smoothed combinations then you should flip the system parameter RESTART_SMOOTHING_FROM_SAVEPOINT to N before re-executing the script.

Supplier and Container Scaling

The following sections describe the process of Supplier and Container Scaling.

Supplier Scaling

Supplier scaling occurs on a daily basis for today up to a user specified scaling horizon. Those days on which the user has specified supplier minimums are a candidate for supplier scaling.

Supplier scaling begins by summing the planned warehouse POs for a release date (also called the order date) and grouping of Suppliers, SKUs, and Warehouses called a

Scaling Group. If the summed PO totals are at least as large as the defined minimums the minimums are considered satisfied for the release date.

If the summed PO totals are less than some of the minimums the system attempts to move POs planned from future release dates to the current release date until the minimums are met.

This process is repeated sequentially for each release date starting from today up-to a user specified supplier scaling horizon in increasing order of release dates.

Container Scaling

Container Scaling (CS) occurs on a daily basis for today up to a user specified scaling horizon. If Supplier Minimum Scaling (SMS) is turned ON then SMS happens before container scaling on any given release date. Those days on which the user has specified container dimension constraints are a candidate for container scaling.

Container scaling first places order quantities into containers according to fit; at the same time attempting to group orders for a particular warehouse into the same containers. An order quantity does not fit in a container if adding the quantity to the existing total would exceed a container maximum constraint. When all orders are placed in a container those containers whose totals do not meet a user specified minimum is subject to scaling. The system attempts to move PO quantities planned for future release dates to the current release until a container minimum has been met.

This process is repeated sequentially for each release date.

Note: For complete details on Container Scaling, refer to the *Oracle Retail Advanced Inventory Planning Order Management User Guide*.

Technical Details

The scaling logic first retrieves the supplier scaling and container scaling system parameters to determine whether either module is enabled at the global level. If not, the script completes successfully having done nothing.

Next, data is pulled together into the driving table, SCALING_GROUP_ORDER_MAP. Yesterday's data is deleted from the table before rebuilding the table for the current run.

Finally, scaling is performed. The process spawns multiple threads to perform the operation over the SCALING_GROUP_ORDER_MAP table partitions. A record is logged in the RESTART_SCALING table each time scaling is completed for a Scaling Group, Release Date, and scaling module (supplier scaling or container scaling). The record aids restart recovery and prevent re-scaling orders which, in certain circumstances, could produce inaccurate results.

Script Name: scale_order.sh

Period to Scale: [releasing_today | releasing_in_future | releasing_any_day].

releasing_today: Scale only today and exit. This allows the critical release process to execute before scaling subsequent forecast days.

releasing_in_future: Scale all forecast days starting from tomorrow. This can be used after releasing_today.

releasing_any_date: Scale all days starting from today through the scaling horizon.

Note: When separating the scaling execution by time period using the releasing_today and releasing_in_future parameters it is critical that the releasing_today script call is made before releasing_in_future.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch. Recovery is built into the scaling logic such that scaling is not repeated for a completed Scaling Group/release date/scaling module (supplier scaling or container scaling).

Partial Pallet Rounding

If partial pallet rounding is turned on, then each order line is rounded before aggregating for comparing against the warehouse receiving capacity or Supplier minimum or Container Scaling constraints.

By default rounding is turned off in system parameter, PARTIAL_PALLET_ROUNDING_FLAG. When moving orders in scaling or smoothing with rounding turned on, the source side rounded total is decreased by an amount equal to the difference of rounded order total before and after the move.

When moving orders in scaling with rounding turned on, the destination side rounded total is increased by either adding the rounded move quantity when no matching order exists on destination side or by adding the difference of rounded order total after and before the move when a matching order exists on destination side.

When loading orders into container, each partial pallet is counted as full pallet towards the container capacity. Please note that rounding is only effective when the constraints are set in terms of pallets and are applicable in smoothing or scaling.

Post-Scaling

When scaling is completed for all days (today and future release days) the modified order quantities and executed Supplier Purchase Quantities (SPQ) are sent to the AIP RPAS platform to be reflected in the WRP workbooks and WRP alert calculations.

Note: The modified order quantities are not re-reconciled.

The RETL exports to be executed are contained in the export_scale.config file. Once the exports are complete they are archived and readied for moving to AIP RPAS.

Script Name: post_scale.sh

This script exports modified warehouse POs as a result of smoothing and scaling and executed SPQ quantities to AIP-RPAS.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.

3. Restart the script.

Merge Purchase Orders

Prior to releasing warehouse purchase orders they must be moved from an intermediate table to the final warehouse order table. A parameter passed into the executing script controls which day or days' orders are copied. This step first clears out unreleased warehouse purchase orders from the NON_CONTENTS_ORDER table. These are yesterday's forecast orders. Only orders that fall between the specified time period are deleted. Then, the new purchase orders are copied from the WAREHOUSE_PURCHASE_ORDER table to the NON_CONTENTS_ORDER_TABLE. Only orders with an order quantity greater than 0 are copied.

Script Name: merge_order.sh

This script moves scaled into-warehouse POs to the final database table for Release.

Input Parameters

Following are the input parameters for the merge_order.sh script.

Release Period: [releasing_today | releasing_in_future | releasing_any_day]. This is a required parameter.

Note: This script is used only for warehouse purchase orders. All other orders and transfers do not require this script to be run before releasing.

Parameter	Action
releasing_today	Clears out unreleased orders with a release date of today and copies new orders with a release date of today to the NON_CONTENTS_ORDER table.
releasing_in_future	Clears out unreleased orders with a release date greater than today and copies new orders, with a release date greater than today, to the NON_CONTENTS_ORDER table.
releasing_any_day	Clears out all unreleased orders and copies the new orders, having any release date, to the NON_CONTENTS_ORDER table.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the script.

Release Orders to RMS

The following sections describe the process of Release Orders to RMS.

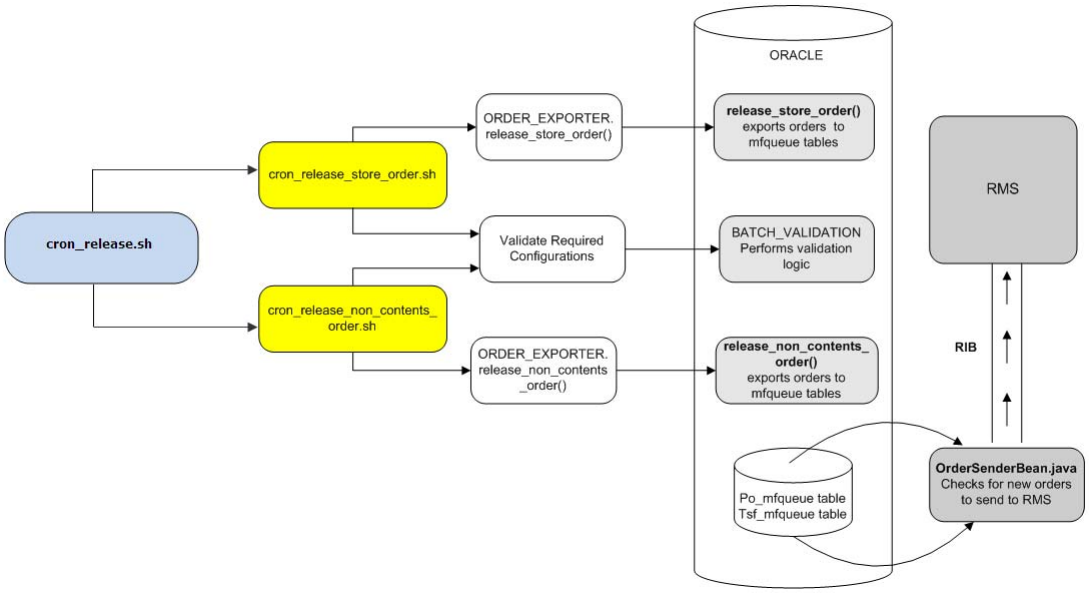
Order Release Overview

After the nightly RPAS batch run, the planned store and warehouse orders are extracted to a set of flat files (.dat data files) that are loaded into AIP Oracle. Similarly

after the Intra-day replanned store orders are extracted to a set of .dat files, they are loaded into AIP Oracle. Order Release batch determines which orders must be released to the order execution system (RMS). Overnight orders that are a candidate for release must have a release date of today. Intra-day orders that are candidate for release must have a release date of today and release wave equal to the given release wave that is passed into the batch release script.The release date is determined by the source lead time. Additionally, the destination of a transfer must have a warehouse-chamber status of either Release or Closing Down. Once these conditions are met, the order is assigned either a purchase order number or a transfer number. The purchase order or transfer is then released to RMS where the SKU, destination, and order quantity are communicated to the supplier or warehouse source.

The released orders and forecast purchase orders are also visible from the OM Online screens. You are able to perform an early release or modify the order quantity, the delivery date, and the destination of a purchase order. In case of Intra-day release if a future release wave is manually released, it's release wave is changed to null indicating manual release. These user-entered modifications must also be communicated to RMS to be executed by the sender of the order.

Figure 7–3 Order Release Flow



Release Orders

An overnight batch release of AIP RPAS generated orders and transfers is performed once in the morning before the online users resume their daily screen activity. All orders and transfers scheduled for release today and are not set to be released in any release wave are communicated to RMS. The orders are assigned a purchase order number or transfer number, and the order status are then set to Open.

The Intra-day batch release is kind of delayed release of into-store orders with the option of replanning prior to release. An order for SKU/store set to be released in a selected release wave can be replanned in prior release waves. Functionally each replanned order replaces the previous order. Technically, the orders are not physically deleted just that they are identified and updated as replanned. At the time of release,

only those orders that are marked to be release on the given release wave and are not replanned (not deleted) are released.

There are some rules about the order in which release waves can be sequenced. The release wave number denotes a nominal time from 0 to 23 where 0 indicates midnight and 23 indicates night 11:00 PM. When configuring the waves, gaps are allowed. For example, you can set up waves such as 2, 5, 8, and 10. Once configured, the waves must be executed in order of increasing wave numbers. Also a wave execution cannot be jumped over to execute a later wave. Wave execution refers to both loading and releasing orders for a particular wave. Overnight load and overnight release always take place before any Intra-day load or Intra-day release.

Technical Details

This process releases store purchase orders, store transfers, warehouse transfers, and warehouse purchase orders to the merchandising system. The order information is written to a set of staging tables. The records on the staging tables provide detail about the order items and the type of action that must be performed by the RMS message subscription logic so that the RMS order is in sync with the AIP order. An Enterprise Java Bean called OrderSenderBean constantly polls the table to find new records. It then reads the details from the staging tables and the base order tables to generate the messages sent to RMS through the RIB.

Script Name: cron_release.sh

This is a wrapper shell script to batch release Purchase Orders and transfers from AIP Oracle tables.

Input Parameters

Following are the input parameters for the cron_release.sh script.

- Order Type (o):[transfer | purchase | all]. This is a required parameter.

Parameter	Action
-o transfer	Releases transfers only.
-o purchase	Releases purchase orders only.
-o all	Releases both transfers and purchase orders.

- Destination Type (d): [store | warehouse | all]. This is a required parameter.

Parameter	Action
-d store	Releases into-store orders only.
-d warehouse	Releases into-warehouse orders only.
-d all	Releases both into-store and into-warehouse orders.

- Release Wave (w): [A number between 0 and 23]. This is an optional parameter that applies to destination_type store orders only. This is used to release Intra-day orders for a given release wave.

  Depending upon the order type, destination_type and release_wave, parameter values passed into this script the desired combination of orders can be released. This script together with the cron_import_order.sh script can be repeated with different parameters to load and release the orders as dictated by time sensitivity sequentially by time precedence.

Example

For many retailers releasing overnight transfers is more time critical than releasing overnight purchase orders. The following example shows an execution order of scripts that can be followed to load and release overnight transfers before overnight purchase orders.

1. Load and release all transfers first.

cron_import_order.sh	-o transfer -d all

2. Next, load and release all overnight POs.

cron_release.sh	-o purchase -d all
cron_release.sh	-o purchase -d all

3. Execute post release and post-critical processing - cron_post_release.sh -o all -d all.

Processing Steps

This script calls the following scripts depending on the value of the destination type parameter passed into it:

- scripts/cron_release_store_order.sh
- scripts/cron_release_non_contents_order.sh

Release Store Orders

This process is executed by cron_release.sh when the dest_type parameter is store or all. When called for releasing overnight orders, release wave parameter is *overnight*. When called for releasing Intra-day orders, release wave parameter is a number between 0 and 23. This process releases store transfers and purchase orders to the merchandising system. Order line items are grouped together and assigned a purchase order number or transfer number, and the order status is then set to Open. The order information is written to a set of staging tables. The records on the staging tables provide detail about the order items and the type of action that must be performed by the RMS message subscription logic so that the RMS order is in sync with the AIP order. An Enterprise Java Bean called OrderSenderBean constantly polls the table to find new records. It then reads the details from the staging tables and the base order tables in order to generate the messages sent to RMS through the RIB.

Technically into-store purchase orders and into-store transfers follow the UPDATE model of release process that is based on assigning order_number and changing order_status using an UPDATE or MERGE statement on order lines already present in main table.

Script Name: cron_release_store_order.sh

The script calls the following PL/SQL wrapper procedure in order to perform the release of store purchase orders and/or transfers. Also listed are some of the important pre-release validation checks called from wrapper procedure.

Note: Users should avoid executing this script directly. Instead the wrapper script, cron_release.sh should be executed.

Procedure of package:	Script
ORDER_EXPORTER	release_store_order

Check for Existence of Order Definitions

Checks to see if order definitions exist.

Causes a batch failure if order definitions are missing. The order definitions are used when assigning order numbers during order release.

Check for Existence of RMS/AIP SKU Mappings

Checks to see if RMS/AIP SKU mappings exist for all SKU/pack sizes.

Orders for a particular SKU/pack size cannot be released to RMS if the mapping is missing.

The validation failure indicator is set to Y on the STORE_ORDER table and an alert is created for the user to view in DM Online.

This does not cause the release process to halt.

The remaining SKU/pack sizes with valid mappings are released.

Check for 0 order quantity

Sets VALIDATION_FAILURE_IND to Y if order quantity is 0.

This prevents the order from being released to RMS.

Restart/Recovery Steps for Releasing Store Orders

In case of failure/errors while running cron_release_store_order.sh through cron_release.sh script, perform the following action:

Review the log file, which is located in the logs folder of the directory specified for the Oracle LOGHOME environment variable.

Error Indicating that the Maximum Transfer or Purchase Order Number was Reached

If you receive an error indicating that the maximum transfer or purchase order number was reached, perform the following steps.

> **Note:** This error means that the system ran out of available order numbers to assign to the orders that are ready to be released. Purchase Order number recycling happens automatically if a custom RMS purge script captures the purged purchase orders and sends them to AIP.

1. Ensure that the custom rmse_order_purge.dat file is being received and processed once the AIP purchase order is purged from RMS.
2. To immediately address the issue, the purchase order numbers and transfer numbers in RMS need to be compared against the range of numbers specified for AIP in the ORDER_NUMBER table.
   - Order_type *P* identifies the AIP number range for purchase orders.
   - Order_type *T* identifies the AIP transfer number range.

3. If a sufficient range of values can be found, the ORDER_NUMBER current_value for the order type can be updated to match the smallest available number. Restart cron_release_store_order.sh from cron_release.sh.

   Note: Purchase order numbers and transfer number ranges may be allocated to many different systems. All of these systems feed orders into RMS, but RMS will not accept duplicate order numbers. Therefore, it is not recommended that you update the ORDER_NUMBER low_value or high_value without a thorough analysis of the entire system.

   If you receive an error indicating that the order definitions check failed, insert the missing rows and restart cron_release_store_order.sh from cron_release.sh.

   Note: This error means that rows were removed from the ORDER_DEFINITION table. This table must contain four rows, one for each destination type and order type combination.

Table 7–2 shows the only order definition configuration supported in AIP.

Table 7–2 AIP Supported Order Definition Configuration 1

DEST_TYPE	ORDER_TYPE	USE_SOURCE	USE_COMMODITY	USE_PACK_SIZE	USE_DEST	USE_DELIVERY_DATE
S	T	Y	N	N	Y	Y
W	T	Y	N	N	Y	Y
S	P	Y	Y	Y	Y	Y
W	P	Y	Y	Y	Y	Y

An error from the orderExporter package indicates that there is an error in the release of orders.

Dependencies

The ability to release orders to RMS depends upon having the AIP RPAS Replenishment Planning batch generated orders loaded in the Oracle database. Therefore, the script should not run until the corresponding import is complete.

Release Warehouse Orders

This process is executed by cron_release.sh when the dest_type parameter is warehouse or all. A batch release of warehouse orders are performed once in the morning before the online users resume their daily activities. Order line items are grouped together and assigned a purchase order number or transfer number, and the order status is then set to Open. All warehouse orders scheduled for release today are communicated to RMS.

The order information is written to a set of staging tables. The records on the staging tables provide detail about the order items and the type of action that must be performed by the RMS message subscription logic so that the RMS order is in sync with the AIP order. An Enterprise Java Bean called OrderSenderBean constantly polls

the table to find new records. It then reads the details from the staging tables and the base order tables to generate the messages sent to RMS through the RIB.

Technically, into-warehouse purchase orders follow the UPDATE model of release process while the into-warehouse transfers follow the INSERT model of release process. The update model of release is based on assigning order_number and changing order_status using an UPDATE or MERGE statement on order lines already present in main table. The insert model is based on assigning order_number as part of INSERT statement when inserting order lines into the main table. Functionally both model of release are equivalent.

Script Name: cron_release_non_contents_order.sh script

The script calls the following PL/SQL wrapper procedure in order to perform the release of warehouse purchase orders and/or transfers. Also listed are some of the important pre-release validation checks called from wrapper procedure.

Note: This script should not be called directly, instead the wrapper script cron_release.sh should be executed.

Procedure of package:	Script
ORDER_EXPORTER	release_non_contents_order

Check for Existence of Order Definitions

Checks to see if order definitions exist.

Causes a batch failure if order definitions are missing. The order definitions are used when assigning order numbers during order release.

Check for Existence of AIP/RMS SKU Mappings

Checks to see if RMS/AIP SKU mappings exist for all SKU/pack sizes.

Orders for a particular SKU/pack size cannot be released to RMS if the mapping is missing.

The validation failure indicator is set to Y on the NON_CONTENTS_ORDER table and an alert is created for the user to view in DM Online.

This does not cause the release process to halt.

The remaining SKU/pack sizes with valid mappings are released.

Check for 0 order quantity and chamber status

Sets VALIDATION_FAILURE_IND to Y if order quantity is 0 or if the destination chamber does not have the status of Release or Closing Down. This prevents the order from being released to RMS.

Restart/Recovery Steps for Releasing Into-warehouse Orders

In case of failure/errors while running cron_release_non_contents_order.sh through cron_release.sh script, perform the following action. Review the log file, which is located in the logs folder of the directory specified for the Oracle LOGHOME environment variable.

Error Indicating that the Maximum Transfer or Purchase Order Number was Reached

If you receive an error indicating that the maximum transfer or purchase order number was reached, perform the following:

> **Note:** This error means that the system ran out of available order numbers to assign to the orders that are ready to be released. Purchase Order number recycling happens automatically if a custom RMS purge script captures the purchase orders and sends them to AIP.

1. Ensure that the custom rmse_order_purge.dat file is being received and processed once the AIP purchase order is purged from RMS.
2. To immediately address the issue, the purchase order numbers and transfer numbers in RMS need to be compared against the range of numbers specified for AIP in the ORDER_NUMBER table.
   - Order_type P identifies the AIP number range for purchase orders.
   - Order_type T identifies the AIP transfer number range.
3. If a sufficient range of values can be found, the ORDER_NUMBER current_value for the order type can be updated to match the smallest available number. Restart cron_release_non_contents_order.sh from cron_release.sh.

> **Note:** Purchase order numbers and transfer number ranges may be allocated to many different systems. All of these systems feed orders into RMS, which will not accept duplicate order numbers. Therefore, it is not recommended that you update the ORDER_NUMBER low_value or high_value without a thorough analysis of the entire system.

If you receive an error indicating that the order definitions check failed, insert the missing rows and restart cron_release_store_order.sh from cron_release.sh.

> **Note:** This error means that rows were removed from the ORDER_DEFINITION table. This table must contain four rows, one for each destination type and order type combination.

Table 7–2 shows the only order definition configuration supported in AIP.

An error from the orderExporter package indicates an error in the release of orders.

Dependencies

The ability to release orders to RMS depends on having the OM-generated orders loaded in the Oracle database. Therefore, the script should not run until the corresponding import is completed.

Post Release

After all the desired orders are released, the post release process should be executed. It is recommended that this script be executed anytime after the overnight release but before OM online users start their daily activities. This step does following:

- Identifies overdue orders.

- Updates order history of into-warehouse orders.
- Refreshes forecast transfers in the non_contents_order table.
- It removes previously loaded forecast transfers in the non_contents_order table and then copies new forecast transfers i_non_contents_transfer into the main tables.
- It signals that users can now start making changes or create new data in Online screens.

Technical Details

The structure of cron_post_release.sh is very similar to the structure of the cron_release.sh script. The scripts take two input parameters, order_type and dest_type, to control the order type and destination types that are processed in any given execution.

Post release is a required post-critical process that should be executed after critical overnight release process is over and before online user activity starts.

Script Name: cron_post_release.sh

This is a wrapper shell script for post release activities performed after cron_release is completed.

Input Parameters

Following are the input parameters for the cron_post_release.sh script.

- Order Type (o): [transfer | purchase | all] This is a required parameter.

Parameter	Action
-o transfer	Executes post-release steps of transfers only.
-o purchase	Executes post-release steps of purchase orders only.
-o all	Executes post-release steps of both purchase orders and transfers.

- Destination Type (d): [warehouse | store | all]. This is a required parameter.

Parameter	Action
-d warehouse	Executes post-release steps of into-warehouse orders only.
-d store	Executes post-release steps of into-store orders only.
-d all	Executes post-release steps of both into-warehouse and into-store orders.

Processing Steps

This script calls the following scripts depending on the value of the destination type parameter passed into it:

- scripts/post_release_store_order.sh
- scripts/post_release_warehouse_order.sh
- scripts/post_release_cross_dock_update.sh

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the script.

Purge Closed Orders in AIP Online

Closed orders are purged on a daily basis to maximize response time of database accesses. The closed order status must be provided to AIP from RMS or the external order management system. Closed orders are purged after a period of time which is configurable by the system administrators. The partitions in store_order and non_contents_order table that become empty as a result of purging closed orders are dropped at the end of order purge process. Then at the end all empty sub-partitions are truncated to release the unused allocated tablespace back to Oracle.

Technical Details

The structure of order purging is also similar to the structure of the cron_release.sh script. The scripts take two input parameters, order_type and dest_type, to control the order type and destination types that are processed in any given execution.

Purging of orders is not a critical process and so it can be scheduled to run in any non-critical time window of the day. However it is recommended that cron_purge is scheduled to run daily to prevent build up of closed orders in the tables.

Script Name: cron_purge.sh

This is a wrapper shell script for purging closed Purchase Orders and closed transfers from AIP Oracle tables.

Input Parameters

Following are the input parameters for the cron_purge.sh script.

- Order Type (o): [transfer | purchase | all] This is a required parameter.

Parameter	Action
-o transfer	Purges transfers only.
-o purchase	Purges purchase orders only.
-o all	Purges both purchase orders and transfers.

- Destination Type (d): [warehouse | store | all]. This is a required parameter.

Parameter	Action
-d warehouse	Purges into-warehouse orders only.
-d store	Purges into-store orders only.
-d all	Purges both into-warehouse and into-store orders.

Processing Steps

The script cron_purge.sh calls the following scripts depending on the value of the destination type parameter passed into it.

Script	Action
cron_purge_store_order.sh	Purges Store Orders
cron_purge_non_contents_order.sh	Purges Warehouse Orders

Purge Store Orders

The following sections describe the process of purging store orders.

Technical Details

This logic purges the closed orders from the STORE_ORDER table that have exceeded their purge age. The purge age is the PURGE_PERIOD (in days) set in the ORDER_PURGE_PERIOD table. The order age is the difference between the timestamp when the order was closed and today. It is recommended that closed store orders are purged by calling the wrapper script cron_purge.sh instead of calling the script cron_purge_store_order.sh directly.

Restart/Recovery Steps for Purging Closed Store Orders

The purging of closed orders is a maintenance task that is not essential to the successful completion of the batch; though overtime it is essential for maintaining manageable table sizes.

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the script.

Purge Warehouse Orders

The following sections describe the process of purging warehouse orders.

Technical Details

This logic purges the closed orders from the NON_CONTENTS_ORDER table that have exceeded their purge age. The purge age is the PURGE_PERIOD (in days) set in the ORDER_PURGE_PERIOD table. The order age is the difference between the timestamp when the order was closed and today. It is recommended that closed warehouse orders are purged by calling the wrapper script cron_purge.sh instead of calling the script cron_purge_non_contents_order.sh directly.

Restart/Recovery Steps for Purging Closed Warehouse Orders

The purging of closed orders is a maintenance task that is not essential to the successful completion of the batch; though overtime it is essential for maintaining manageable table sizes.

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the script.

Purge PLSQL Logs

If PLSQL logging is turned on, then log messages are written to a table called plsql_ log. In order to avoid build up of excessive log, it is recommended that old log messages are regularly purged in non-critical time. A shell script called purge_log.sh is provided in integration home of AIP Oracle batch scripts to do the purging. There is no input parameter for this script.

Logging can be turned on by setting the following two system parameters:

1. PLSQL_LOG_LEVEL: This indicates PLSQL logging level. Valid values are DEBUG, INFORMATION (default), ERROR and NONE. While DEBUG level is the most informative, ERROR level is the least informative logging. Logging level NONE results in no logging at all.
2. PLSQL_LOG_TARGETS: This indicates where the PLSQL logs are written. Valid values are FILES_ONLY (for BSA log files), TABLES_ONLY (for logging table) and FILES_AND_TABLES (for both, also the default).

Another system parameter, PLSQL_LOG_PURGE_PERIOD, is used to control the length (in number of log days) of log retention. Records that have logging VDATE <= (current VDATE - PLSQL_LOG_PURGE_PERIOD) are purged when purge_log.sh is executed. Default value of this parameter is seven days.

8

AIP Interval Batch Scripts

This chapter describes the AIP interval batch scripts.

Integration with Replenishment Optimization

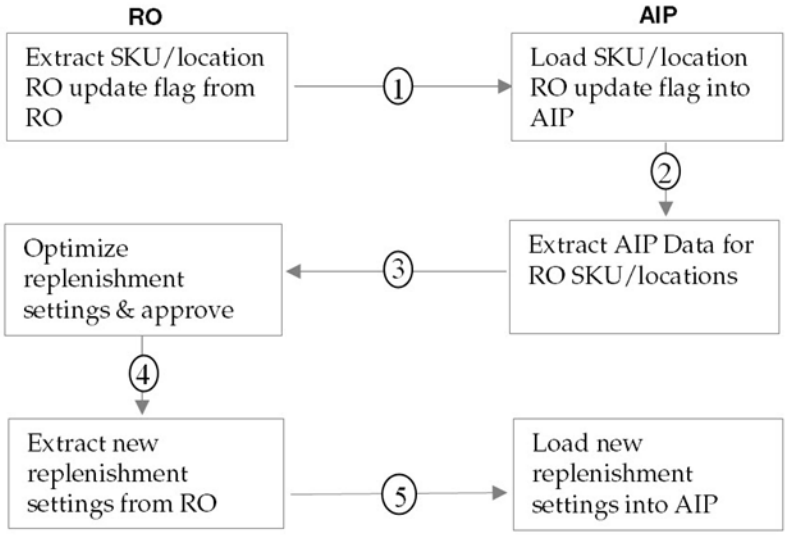
Replenishment Optimization (RO) suggests the most appropriate replenishment method and thresholds to use based on the business goals set in RO, and the supply-chain. The supply-chain information is provided to RO by AIP. The optimized settings should then be used in AIP when replenishment planning.

AIP provides a number of control parameters to allow flexible control over the integration.

- The frequency of import and export from AIP is controlled by the frequency of scheduled tasks. Depending on the Retailer's business strategy, new SKU introduction, and so forth, integration updates could occur weekly, monthly, quarterly, or less as needed.
- The **Replenishment Optimization Location Types** value can be used to limit the locations processed to only stores or only warehouses.
- The **Replenishment Optimization Update for Stores/Warehouses** measures can be used to limit the data set that is sent to RO. The measures are editable within AIP as well as loadable from RO.
- The **Replenishment Optimization Start Date** measure is available for specifying a date to begin pulling data for extract. This helps to eliminate the time-phased disparity between AIP and an optimization system. The Retailer can run the optimization for the supply-chain ahead of when the supply-chain changes. In the workbooks the standard hierarchy rollups are available for mass selecting SKUs and locations to set the date. The Replicate spread method maybe be used to set the same date (that is, start of quarter) at an aggregate level. This value is optional. The batch run date is used when no value is specified.

Figure 8–1 illustrates the flow of data between AIP and RO.

Figure 8–1 Flow of Data between AIP and RO



AIP Import From RO

AIP imports the new, optimized replenishment settings used for planning. Because of the critical nature of the data being loaded from RO, AIP ensures the integrity of the data by validating that every SKU/location replenishment method has corresponding thresholds and vice versa. AIP does not load a SKU/location replenishment method which is missing one or more of the needed thresholds. Similarly, AIP does not load SKU/location parameters for which no replenishment method has been provided. If one or more values are missing for a SKU/location the load logs an error message and continues loading subsequent SKU/location values.

This batch step is not required to operate AIP replenishment. This step is only for those Retailers integrating with RO or who wish to import the replenishment method data from an external system. Alternatively, users can specify replenishment method data in the SRP and WRP workbooks.

Script Call

aip_import_from_ro.sh

Functional Overview

This script imports optimized replenishment data provided by RO into AIP. The approved, or working, settings are extracted from RO for loading into AIP. The replenishment method and corresponding thresholds from RO are loaded at a SKU/destination/effective date level.

Technical Details

The following files are generated by RO and are optional. The file contents are designed in a way that each row contains all required and necessary data elements. Each file is first validated and then loaded into AIP measures.

File Name	Description
STR_MINMAX	Contains Store minmax replenishment parameters.

File Name	Description
STR_DYNAMIC	Contains Store dynamic replenishment parameters.
STR_TIMESUPPLY	Contains Store Timesupply replenishment parameters.
STR_HYBRID	Contains Store Hybrid replenishment parameters.
STR_POISSON	Contains Store Poisson replenishment parameters.
STR_MINSS	Contains Store Minimum safety stock data.
STR_ROUPDATE	Contains information of intersections for which export/ import need to occur for stores.
WH_MINMAX	Contains Warehouse minmax replenishment parameters.
WH_DYNAMIC	Contains Warehouse dynamic replenishment parameters.
WH_TIMESUPPLY	Contains Warehouse Timesupply replenishment parameters.
WH_HYBRID	Contains Warehouse Hybrid replenishment parameters.
WH_POISSON	Contains Warehouse Poisson replenishment parameters.
WH_MINSS	Contains Warehouse Minimum safety stock data.
WH_ROUPDATE	Contains information of intersections for which export/ import need to occur for warehouses.

The details of data elements and their formats have been detailed in the *Oracle Retail Advanced Inventory Planning Implementation Guide*.

The batch creates a <filename>.error file for bad records found in each of the previous mentioned files. The import batch processes one file at a time for available files present in the $RO_INPUT directory.

Each file is validated and then loaded in the system. Standard AIP logs are created through BSA.

Prerequisites

Files must be copied or FTP'ed to the $RO_INPUT directory and unpacked by a Retailer-created scheduled task before executing this script.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the import of data by executing aip_import_from_ro.sh. The files already processed by batch need not be reloaded.

AIP Export to RO

The following sections describe the AIP Export to RO.

Script Call

aip_export_to_ro.sh

Functional Overview

This script exports lead time, review time and pack size information from AIP into pre- formatted/csv flat files for RO. This batch step is not required to operate AIP replenishment. This step is only for those Retailers integrating with RO or who wish to export the data to an external system.

Lead Time

The lead time (order cycle) pattern generally always contains the same lead time on all days that have a lead time, or the lead time increases for the weekend. Therefore it is expected that the most common lead time is found during the business week.

Review Time

Review Time is the number of days until the next possible receipt + availability lead time. The review time is a key factor in determining the minimum amount of projected stock that should be available until the point of ordering. The projected available inventory needs to cover the projected need until the next earliest possible receipt. Because review time can change daily or cyclically, in order to avoid stock outs, the minimum available inventory must cover the longest review time.

Ordering Pack Size

The Ordering pack size is the preferred pack size of a SKU that should be ordered from a source to destination.

Technical Details

The following process describes the AIP Export to RO.

1. The batch script runs the rule group RoExtract on each local domain and populates temporary measures.
2. The data in the temporary measures is extracted in the flat files STR_AIP_DATA and WH_AIP_DATA. The file contents are designed in a way such that each row within the file contains all required data elements.

File Name	Description
STR_AIP_DATA	Contains Store lead time, review time and pack size data.
WH_AIP_DATA	Contains warehouse lead time, review time and pack size data.

The details of data elements and their formats have been detailed in the *Oracle Retail Advanced Inventory Planning Implementation Guide*.

The data extracted to the files is limited by:

- The Replenishment Optimization Location Types setting which indicates the location types-stores and/or warehouses-which are integrated with RO.
- The replenishment optimization update flag for stores or warehouses that is set at a SKU/location intersection and indicates which SKU/stores and/or SKU/warehouses should be extracted for updating and optimization in RO.

Note: For each location type that is specified in the **Replenishment Optimization Location Types** measure the RO update flag is retrieved. If no SKU/location combinations are flagged for update of a particular location type then ALL SKU/location combinations of that location type are extracted.

---

3. The batch script clears all the temporary measures.
4. Files generated are placed at the location configured in the environment variable $RO_OUTPUT. The outbound files from the AIP RPAS server should be retrieved by FTP using a job scheduling application.

Prerequisites

When running a stores extract the STR_ROUPDATE file should be loaded from RO first to indicate which SKU/stores are being optimized in the next Optimization run. Optionally, AIP users may manage this setting in an AIP Workbook.

When running a warehouse extract the WH_ROUPDATE file must be loaded from RO first to indicate which SKU/warehouses are being optimized in the next optimization run.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the export of data from AIP by restarting the script aip_export_to_ro.sh.

AIP Data Purge

Hierarchy and historical data is purged from Oracle database tables using aip_purge.sh script. Hierarchy data that is not loaded into Oracle table for the specified number of days or historical data where the end date is in the past is deleted from the tables. In addition to this, data that has become invalid due to invalid or missing relationships is also deleted from tables. Hierarchy data left after purging is exported to AIP-RPAS so that AIP-RPAS can also perform the necessary purging/cleanup.

AIP-RPAS loads the hierarchy data received from AIP-Oracle during a purging interface with a purge age set to 0, indicating that only the data in the load files remain in the AIP domain's hierarchy arrays at the completion of the load. In addition to the hierarchy data purging, measures whose values are positions along hierarchy dimensions are considered to ensure that no purged hierarchy positions remain as values.

The AIP-Oracle and AIP-RPAS procedures described in this section are not a part of the regular AIP batch processes. They are initiated separately and are controlled by different scripts.

AIP Oracle Data Purge

This section describes the process for the AIP Oracle data purge.

Script Call

aip_purge.sh

Input Parameters

None.

Technical Details

This is not a critical process and so it can be scheduled to run in any non-critical time window of the day. Also it does not need to run daily. It can be scheduled to run once a

week or once a month or less frequently. Due to the functional dependencies involved in purging data, certain tables must be purged first before other tables. Also due to technical dependencies (like referential key constraints) child tables must be purged before parent tables. Table purge_parameters is used to configure the purge age of hierarchical data tables.

The script purges data in groups of parent tables. Each purge group includes several parent tables that are purged in parallel. Groups themselves are purged in sequence. For example, first group to be purged includes following parent tables: supplier, warehouse, store, commodity_pack_size, chamber, historic data and on_supply_off_supply. At the end, of purging invalid data in parent tables, the remaining hierarchy positions are exported to AIP on RPAS in a compressed tarred file called purge_hierarchy.tar.Z.

Restart/Recovery

Before any parent table is purged, a log is inserted in a temporary log file which is removed at the end if all tables were successfully purged. If aip_purge.sh script fails in the middle, then following actions can be taken:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the script. Parent tables that were successfully purged are not attempted again.

AIP RPAS Data Purge

The AIP RPAS Data Purging master script, purging_aip_batch.sh, accepts a number of arguments that allow more control over what portion of the batch scripts are run. Step names have been defined, which may also be passed into the script as arguments that define exactly which steps and corresponding scripts should be run.

Usage

The following table provides descriptions of the purging_aip_batch.sh arguments

Argument	Description
-s	Indicates that a starting step is defined. The step at which the purging batch should start must follow this flag. Only one batch step can follow the flag. This flag must be used along with the -e flag and its associated argument.
-e	Indicates that an ending step is defined. The step after which the purging batch should end must follow this flag. Only one batch step can follow the flag. This flag must be used along with the -s flag and its associated parameter.

Steps for Purging_aip_batch.sh

The following are valid steps that can be used with the -s and -e flags. These steps can also be passed as parameters to the purging_aip_batch.sh script, in the form of a list of steps (not flagged with -s and -e). Each step is described in detail in the following section.

Execution Sequence of the AIP RPAS Purging Batch Scripts

Table 8–1 describes the steps in the order that purging_aip_batch.sh execute them.

Table 8–1 Execution Sequence of the AIP RPAS Purging Batch Scripts

Step Name	Script Name	Technical Details	Purging Batch Run
purging_prep_oracle_data	purging_prep_from_ aiporacle.sh	Prepares the export files from AIP Oracle for loading into the AIP RPAS platform.	Yes
purging_process_ hierarchies	purging_process_hierarchies.sh	Pre-processes the hierarchy data exported from AIP Oracle before it is loaded into RPAS domain by inserting positions into labels, removing field separators, and creating stocking point hierarchy load files.	Yes
purging_load_hierarchies	purging_load_hierarchies.sh	Purges and load all hierarchies in the AIP domain.	Yes
purging_process_measures	purging_process_measures.sh	Deletes positional measure values from the domain's measure arrays.	Yes

The command line interface of purging_aip_batch.sh is identical to that of aip_ batch.sh, described previously, except that there is no first-day flag (-f). All the steps of the purging_aip_batch.sh script are intended to be run in the same window of time, in order that the purging is synchronized across both hierarchy and measure data. However the interface to this script is modeled after aip_batch.sh for consistency.

Prepare Purging Data from AIP Oracle Platform

Step Name

purging_prep_oracle_data

Script Call

purging_prep_from_aiporacle.sh

Technical Details

AIP RPAS processes data from AIP Online. AIP Online exports hierarchy information and DM measures which are placed by the user into the AIP RPAS domain, in the $AIPDOMAIN/interface/import directory.

The prep_from_aiponline.sh script is called to process the data files created by the AIP Online export. The data files should be transferred from AIP Online to the export directory by a job scheduling application.

Interval Processing

1. Using prep_files.sh, the archive file purge_hierarchy.tar.Z is unpacked from $AIPDOMAIN/interface/purge/import into $AIPDOMAIN/interface/purge/import/hier.
2. The unpacked files are verified to ensure all the data from AIP Oracle Purging batch is present, as listed in the $AIPDOMAIN/interface/purge/config/purge_ hier_import.config file. The following hierarchies mastered by AIP Oracle Purging are expected to be in purge_hierarchy.tar.Z.

File Name	Explanation
hspl.dat	Supplier Hierarchy
loc.dat	Location Hierarchy
ntwg.dat	Network Group Hierarchy
oltc.dat	Order Lead Time Cycle Hierarchy
ordg.dat	Order Group Hierarchy
proc.dat	Profile Order Cycle Hierarchy
prod.dat	Product Hierarchy
prof.dat	Profile Hierarchy
whse.txt	Warehouse Hierarchy

Restart/Recovery

If this step fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Ensure that the exported data from AIP Oracle exists inside the purge_hierarchy.tar.Z file, and this file is located in $AIPDOMAIN/interface/purge/import.
4. Ensure the contents of purge_hierarchy.tar.Z match those in the $AIPDOMAIN/interface/purge/config/purge_hier_import.config configuration file. All files are required.
5. Restart the purging batch.

Process AIP Oracle Purged Hierarchies

Step Name

purging_process_hierarchies

Script Call

purging_process_hierarchies.sh

Technical Details

This step pre-processes the AIP Oracle purging exported hierarchy data before they are loaded into RPAS domain. The processing consists of inserting positions into labels, removing field separators, and creating stocking point hierarchy data.

Interval Processing

The following describes interval processing.

1. Copy all files listed in $AIPDOMAIN/interface/purge/config/purge_hier_import.config from $AIPDOMAIN/interface/purge/import/hier to $AIPDOMAIN/input.
2. Move clnd.csv.dat, if it exists, from $AIPDOMAIN/interface/purge/import/hier to $AIPDOMAIN/input.

The Source Stocking Point and Destination Stocking Point hierarchies are created in $AIPDOMAIN/input using elements of the Warehouse, Supplier and Location hierarchies.

The positions are prepended to the position labels in the whse, hspl, ssp, and dsp hierarchy data files.

3. Export the product and location hierarchies from the AIP RPAS domain, and reduce them by the SKPS (from product) and STR (from location) keys from the AIP Oracle purging export. Write the outputs to the $AIPDOMAIN/input directory.

Restart/Recovery

If this step fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Ensure that the individual hierarchy data files to be loaded (as specified in the $AIPDOMAIN/interface/purge/config/purge_hier_import.config) exist in the $AIPDOMAIN/interface/purge/import/hier directory.
4. Restart the purging batch.

Load All Purged Hierarchies

Step Name

purging_load_hierarchies

Script Name

purging_load_hierarchies.sh

Technical Details

This step loads the purged hierarchy data processed in the previous steps, using a purge age of 0. The end result is that only the hierarchy data positions in the load files survive the load process. Those RPAS measures whose base intersections include dimensions that have been purged of unneeded positions are re-sized to not include those positions.

Although calendar hierarchy purging is not mastered by AIP Oracle, the client may purge their calendar hierarchy during this step. Either copy a manually constructed clnd.dat containing only those days to be retained, into the $AIPDOMAIN/input directory, or run the RMS calendar extract, process it through the RMS-AIP Transformation scripts to generate a clnd.csv.dat, which is copied to the $AIPDOMAIN/interface/purge/import/hier directory, prior to running purging_load_hiearchies.sh. For details on the RMS-AIP Transformation process, read the chapter on AIP Interfaces and Transformation Scripts in this guide and the *Oracle Retail Advanced Inventory Planning Implementation Guide*.

Interval Processing

Call loadHier RPAS utility with -loadAll and -purgeAge 0 options to load all the processed hierarchy data files into the AIP RPAS domain.

Restart/Recovery

If this step fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Ensure that the individual hierarchy data files to be loaded exist in the $AIPDOMAIN/interface/purge/import/hier directory.
4. Restart the purging batch.

Process AIP RPAS Positional Measures

Step Name

purging_process_measures

Script Call

purging_process_measures.sh

Technical Details

Some AIP RPAS measures contain values corresponding to positions along a dimension. This guide refers to these measures as positional measures and their data as positional values. During the previous load hierarchy step, unused positions are removed from the hierarchy arrays. If any of these measures' positional values are purged positions, the loadHier step does not clean these up. Those measures that have been predetermined to be positional measures are listed in the configuration file $AIPDOMAIN/interface/purge/config/purge_meas_values.config. The configuration file also lists the dimensions in which the measure's value can be a position. This step iterates through each measure in this configuration file and verifies that their values still exist in the hierarchy. If a purged value is detected, it is replaced with the measure's NA Value to keep the hierarchy and the positional values in synchronization.

Interval Processing

Iterate over $AIPDOMAIN/interface/purge/config/purge_meas_values.config and execute in parallel the purgeMeasureValues binary on each measure.

Restart/Recovery

If this step fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the purging batch.

AIP Warehouse Feedback

This script must be called before other scripts in the AIP batch. This ensures that the data exists on the correct date and locations.

For each local domain the Baseline Safety Stock Units Override is calculated. Once this is calculated the Re-planned URP Output measure is cleared for the upcoming batch run.

Once this script has been run it must be run in each successive batch run. To stop running this script, between batch runs both the Baseline Safety Stock Units Override and Re-planned URP Output measures must be cleared.

Script Call

warehouse_feedback.sh

Input Parameters

None.

Restart/Recovery

This script follows the BSA method of logging. The logs are placed in the directory defined by BSA.

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the script.

9

AIP Batch Environment Maintenance

The maintenance of servers and applications is an ongoing necessity. Occasionally, it may be necessary to perform maintenance on the existing AIP RPAS global domain.

Refer to the *Oracle Retail Advanced Inventory Planning Implementation Guide* for information on configuring and partitioning AIP RPAS domains at build time.

Notes on AIP Domain Builds

There are three directory path references inside of an AIP global domain that must be mentioned before discussing domain relocation.

Table 9–1 Directory Path References

globaldomainconfig.xml	The AIP Implementation Guide provides instructions regarding customizing the globaldomainconfig.xml file required for an RPAS domain build using the build_aip_domains.ksh script. This XML file contains the absolute path to all domain components, namely, the master domain path and all local domain paths.
configmeasdata.db	The master domain contains the directory configmeasdata.db, located in $AIPDOMAIN/data directory. This directory contains an array named r_subdomainpath%1 which contains a map from all partitioning dimension positions to the local domains which contain each position.
admin.db	Each local domain contains the directory admin.db, located in each local domain's data directory. This directory contains an array named domain_properties. This array contains the path to the master domain.

It is important that these references to the master domain's path and local domains' paths be synchronized at all times to ensure the global domain's integrity. Following the procedures in this chapter ensures the pointers are correct. Using UNIX mv or cp on the domain build directory (master domain) or sub-directories (local domains) results in a corrupted set of internal links.

In addition, the shell script $RPAS_HOME/bin/aip_env_rpas.sh contains the assignment of the $AIPDOMAIN variable. If the master domain is moved, then the $AIPDOMAIN variable must also be updated to reflect the new path.

The standard domain build procedure described in the AIP Installation Guide and AIP Implementation Guide specifies that the absolute path to each domain component be specified in the globaldomainconfig.xml file prior to running the build_aip_domains.ksh script. As a result, the three references listed in Table 9–1 contain absolute paths at the conclusion of the domain build process.

The globaldomainconfig.xml can be configured such that the master domain and local domains do not reside in the same directory. For example, the local domains can, but do not have to, be subdirectories of the master domain. This is configurable according to the needs of the client to accommodate disk space.

Domain Relocation

There are two supported utilities that can be run on the AIP domain to relocate all or part of the domain from one directory to another directory. Operations that are supported are those which maintain the integrity and correctness of the three domain references listed in the previous section, Notes on AIP Domain Builds.

Moving a Local Domain or a Master Domain

If required, a local domain or the master domain may be moved from one directory to another. The RPAS utility, moveDomain, can be used to accomplish this task. See the Oracle Retail Predictive Application Server documentation for usage information on the moveDomain utility.

Consolidating the Master and Local Domains

The global domain can be consolidated into a directory structure where all local domains are subdirectories of the master domain.

Consolidating the Domains

For example, given master domain path:

```
/files1/AIPm
```

and local domain paths

```
/files2/AIPl-a
```

```
/files3/AIPl-b
```

using the command:

```
copyDomain -d /files1/AIPm
```

results in the following paths:

```
/files1/AIPm/AIPl-a
```

```
/files1/AIPm/AIPl-b
```

By not specifying a target domain destination, the copyDomain utility updates the domain to have relative paths. Copies of spread out local domains are made as subdirectories of the master domain, and the references between the master and local domains are updated.

Note: This operation leaves behind orphan local domains /files2/AIPl-a and /files3/AIPl-b, which are no longer attached to a master domain. These domains should be deleted.

Other Uses of copyDomain

While not elaborated in this Operations guide, copyDomain has other command line options which can be used to copy part of a local domain, translate from UNIX to

Windows, and compress the copied domain. See the usage information for the utility (copyDomain -help) and the RPAS documentation for full usage of this utility.

User Administration

The AIP RPAS domain does not contain any accounts. The domain build process for previous versions of AIP and RPAS created the default accounts of adm and usr during the domain build. These accounts are not created in the domain build process for this release.

For details and instructions on how to create admin and user accounts for the AIP RPAS domain, read the following documentation:

- *Oracle Retail Predictive Application Server Release Notes*
- *Oracle Retail Predictive Application Server Administration Guide* for the description of the usermgr utility in the chapter, *Operational Utilities*
- *Oracle Retail Predictive Application Server Administration Guide* in the chapter, *Security and User Administration*

Position Reclassification Process

On occasion it is understood that clients may wish to reclassify positions. Reclassification is changing a dimension position's rollup position from one position to another. For example, SKU 30 might roll up to subclass 15, but as a result of a reclassification, the SKU would instead roll up to subclass 40. AIP supports reclassification at hierarchy load time by loading a hierarchy data file that contains the new rollup. This can happen during calls to the loadHier and reconfigGlobalDomainPartitions RPAS utilities embedded inside the AIP batch scripts.

In order for reclassification through loadHier or reconfigGlobalDomainPartitions to be successful, the client should note the old and new rollup positions in relationship to the local domains of the RPAS global domain. If, by reclassifying a position, the position would roll up into a dimension position located in a different local domain than the former rollup dimension position, then the dimension must be buffered in order for the reclassification to be successful. In addition, all dimensions, below the dimension whose position is reclassified, must be buffered.

The default configuration shipped with AIP (as listed in the hierarchy.xml) contains buffering percentages set to 0% low and 0% high for all dimensions. As a result, buffering is disabled. The affected dimension's buffering percentages must be non-zero for the reclassification from one local domain to another. Therefore, before trying to load hierarchy data that has reclassified positions where the local domain rollups change, please read the RPAS documentation on how to use the dimensionMgr RPAS utility to manually increase buffering percentages on the dimensions whose positions are reclassified. This operation takes place outside of AIP batch.

To continue the example: if the domain is partitioned across subclass, SKU 30 rolls up to subclass 15 in local domain ldom0. Subclass 40 is in local domain ldom1. The reclassification requires SKU 30 to be moved to a new local domain and by default the low and high buffering percentages for all dimensions are 0%. Therefore, the SKU dimension and all dimensions below it must be rebuffered in order for SKU 30 to move from local domain ldom0 to the local domain ldom1 during the reclassification hierarchy load.

Oracle Order Table Partitions

In AIP-Oracle, store_order and non_contents_order are the key tables that store released and forecast orders. These are partitioned in such a way that they can support a big volume of load, release and purge cycle. Although the AIP-Oracle batch processes do the partition maintenance by themselves, there can be an occasional need for manual maintenance on some of the partitions. This section describes the structure and the usage of partitions on these tables.

These two tables are:

- First range-partitioned on release_date column (ship date of order)
- List- subpartitioned on source_type column

Partition Format

In AIP, partition and sub-partition names are Oracle database generated names. Typically partition names are of the format **SYS_PXXXXX** and sub-partition names are of the format **SYS_SUBPXXXXX**.

A default partition named **FIRST_RELEASE_DATE** is provided with each table with a release_date upper bound of January 01, 1970 to act as a seed partition. Partition FIRST_RELEASE_DATE by itself normally remains empty.

Each partition has two subpartitions: A subpartition to store purchase orders and a subpartition to store transfers.

New Range Partitions

In AIP new range partitions are created at run-time by Oracle database whenever new records are inserted. The default partition interval size is 1 release date which means that each partition can have at the most 1 release date specific orders. If there are no orders for a given release date but there are orders for a later release date then there may be a separate partition for the later release date but no additional partition for the given release date that has no orders. This way of creating partitions is done by Oracle 11g database itself.

Upgrading

When upgrading to this version of AIP from a previous release, any existing data in store_order and non_contents_order table is preserved by scripts provided in upgrade patch. However it is still recommended that you backup the existing data in these tables before executing any upgrade script.

Removal

While Oracle 11g database is responsible for run-time partition creation, AIP does run-time partition removal as part of Order Purge process. The Order Purge process first creates a list of distinct release dates for which there are physical orders in the table. It then executes the main purging procedure that physically removes the closed orders that have exceeded their purge age. Then using the distinct release date list, it drops all partitions that are left empty as a result of purging done by main purging procedures.

10

AIP Daily Process

The AIP product suite is designed with an online user interface and an overnight and Intra-day batch process. In the online interfaces, the planners log in to the workbooks, DM Online, or OM online, and then set up the parameters for replenishment across the stores and warehouses. The period of time during the day that the planner can access the online interfaces is called the online day. At the end of the business day, users are locked from making changes to data that is used by overnight batch and the batch process execution starts. The overnight batch process performs all the replenishment plan calculations and generates alerts before the next online business day begins.

AIP Online Day

During the online day, AIP users log in to the SRP, WRP, DM, and OM client interfaces to maintain all the parameters and supply chain information for the replenishment and allocation calculations across the stores and warehouses.

Figure 10–1 AIP Online Process

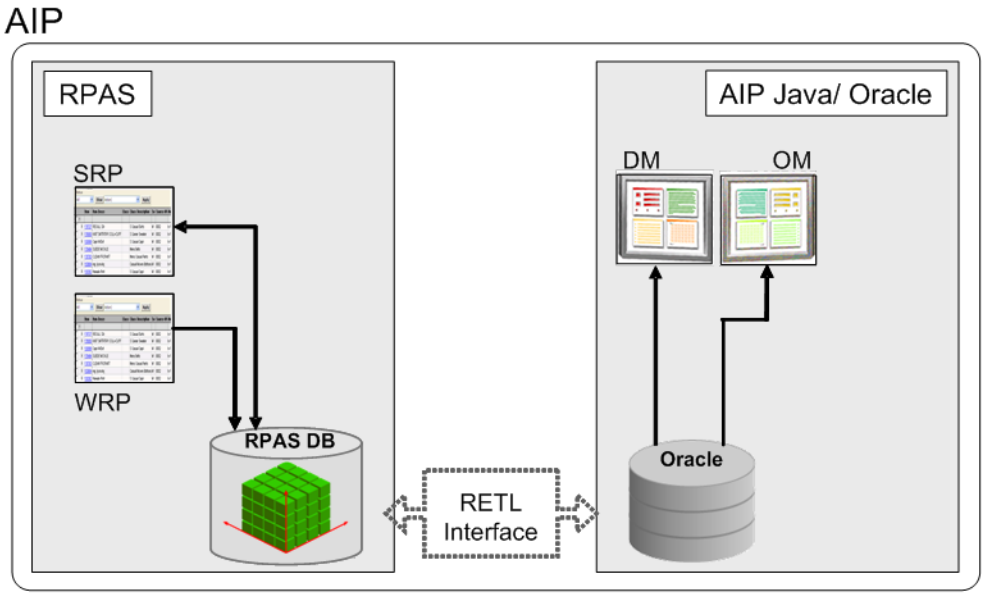


Table 10–1 AIP Acronyms

Acronym	Definition
SRP	Store Replenishment Planning
WRP	Warehouse Replenishment Planning
OM	Order Management
DM	Data Management
RETL	Retail Extract, Transform, and Load

The SRP and WRP users log in to the respective workbooks to maintain the parameters for the calculations, and also to resolve the alerts that were raised during the previous nightly batch process.

Data Management users log in to maintain the supply chain across the warehouses and stores. Order Management users log in to work on the orders created during the nightly batch process. Any modifications to the orders are immediately communicated to RMS.

Once these online users are done with their activities, they can submit all the planning parameters to the AIP-RPAS or Oracle database. Later, during the batch execution process, this data is imported for the replenishment and allocation calculations.

AIP Batch Process

The AIP batch processing takes two paths of execution. They are:

- Initial Batch Run
- Daily Batch Run

Initial Batch Run

After AIP is installed, the RPAS and the Java/Oracle databases must be populated with foundation data before the supply chain setup can be completed. A limited run of the batch scripts is used to accomplish this task on both the RPAS platform and the AIP Java/Oracle platform. Foundation data is first populated on the RPAS platform. Then, the data is copied into the Oracle database. Using this data, the DM user logs into the client application to configure the supply chain. The user must also log in to the AIP workbooks to define the replenishment parameters. Once this is done, the AIP system (across both the platforms) is ready for the daily batch run. More information on the first day process can be found in the *Oracle Retail Advanced Inventory Planning Implementation Guide*.

Daily Batch Run

This batch run is executed on a daily basis. Before executing this batch run, confirm the successful execution of the initial batch run and the DM configurations.

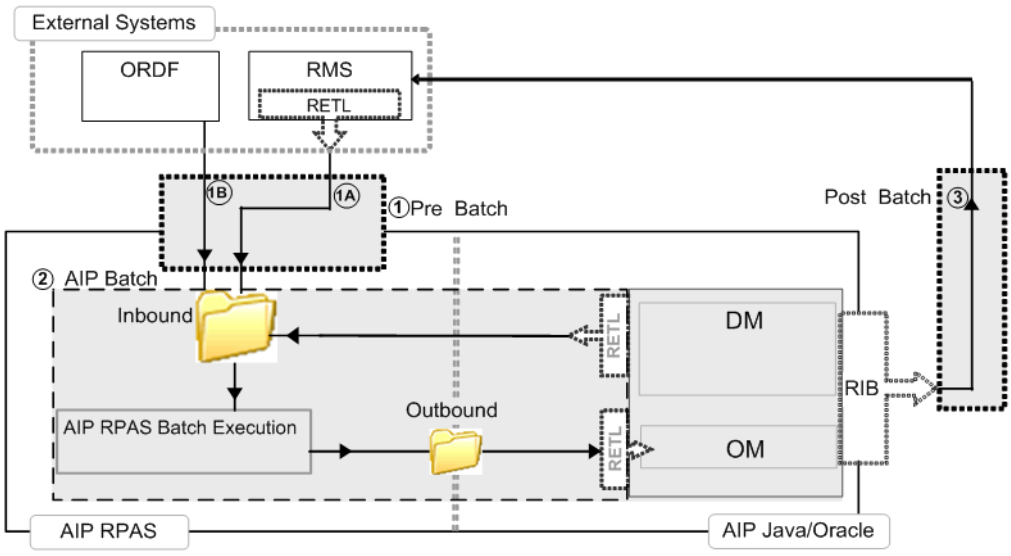
The daily batch run is the process that is outlined in this chapter.

After the online day ends, the daily batch process begins. The entire overnight or Intra-day batch process for AIP is defined in three phases:

Table 10–2 AIP Batch Process

Order	Batch Phase	Description
1	Pre AIP-Batch	Pre AIP batch includes all processes that occur in the integrated Oracle Retail applications (that is, RMS), as well as those processes that occur in custom integrated systems, for the purpose of providing data to AIP. These processes extract the necessary data that is loaded into AIP.
2	AIP Batch	AIP batch is executed in a series of scripted steps that occur on each of the two AIP platforms. To begin, the data required to perform AIP replenishment must be available in a loadable format. This involves extracting data from the Oracle database as well as massaging data files from RMS. Once the data has been loaded the replenishment and replenishment related setup commences. At various points of completion data is moved/extracted from the RPAS database and readied for the Oracle database. This includes data from other systems as well as AIP data created or modified by the AIP on RPAS batch processes. The data is loaded into the Oracle database as it becomes available. During overnight batch, automated supply-chain setup commences in the Oracle database after the merchandise and organizational hierarchies have been loaded. Once replenishment is complete the plan is exported from the RPAS database and loaded into the Oracle database. The overnight plan is passed through the Supplier and Container Scaling modules, when enabled, and then POs and Transfers that have met their lead time are released to RMS or an order execution system. Finally at the end of overnight batch post replenishment activities are performed on the RPAS platform which includes generating receipt plan alerts and automatically building workbooks.
3	Post AIP-Batch	Post AIP batch occurs as soon as the POs and Transfers have completed the AIP release process. The post AIP batch processes must retrieve and load the AIP POs and Transfers that must be executed or communicated to sending and/or receiving entities.

Figure 10–2 AIP Daily Batch Processing Steps



Pre AIP Batch

In the pre AIP batch process, the data for AIP is extracted from the external systems and copied or transferred through FTP into the inbound AIP directory.

The Pre-AIP Batch Process in External Systems

The following table describes this phase of the process as shown in Figure 10–2.

Phase	Description
1A	During the RMS batch process, RETL scripts extract data from the Oracle database. The files must be copied or transferred through FTP into the Inbound directory.
1B	The RDF data for AIP is extracted and loaded into the common directory. During the RDF batch process, the batch scripts are executed to extract information from RDF and place it in the Inbound folder.

AIP Batch

The AIP batch process is executed on both the AIP RPAS and AIP Java/Oracle platforms.

The AIP Batch Process on the RPAS Platform

The following list describes this phase of the process as shown in Figure 10–2.

1. Transforms the RMS data files into AIP loadable format.
2. Runs the AIP batch scripts for all daily batch steps:
   - Formats and loads the RMS data files and other externally generated files from a directory (Inbound folder)
   - Merges and loads the DM online files from a directory
   - Loads the forecast data files from a directory
   - Runs the replenishment and reconciliation logic
   - Exports the AIP RPAS batch output data into a directory (Outbound folder)
   - Loads a modified AIP-scaled order plan
   - Generates alerts
   - Builds SRP and WRP workbooks configured for auto-workbook builds

The AIP Batch Process on the Java/Oracle Platform

Time wise the AIP batch processes that run on the Oracle platform straddle the AIP processes that run on the RPAS platform. Before any of the replenishment planning calculation preparation can begin in AIP RPAS the data must be extracted out of the AIP Oracle database. The cron_export script is used to accomplish this.

After the replenishment plan has been generated the plan is exported to an outbound data directory in the form of orders (purchase orders and transfers). The planned orders, along with the RMS enterprise data, and any DM data that was created or modified in the AIP RPAS batch processes is retrieved and loaded by the AIP Oracle batch process. Use of the cron_import and cron_import_order scripts accomplish this process.

An alternate approach to importing DM data into AIP Oracle can be used to maximize CPU utilization and shorten the nightly batch window. DM data can be imported into

AIP Oracle immediately after it has been exported from AIP RPAS. Specifically, the *cron_import DM* step can be executed immediately after the *export_dm_data* step is complete. For detailed information on these steps, please refer to Chapter 4, "AIP RPAS Batch Processing." This process improves performance by executing the import of DM data at a time when the AIP Oracle server may otherwise be under little or no load.

After the necessary DM data is imported the cron_import script then initiates a number of processes that automatically set up the supply chain for new hierarchy data. These processes also automatically default some supply chain data when missing, and replaces some supply chain data when invalid.

The planned warehouse purchase orders, once loaded, are then passed through a module which pulls-forward future POs in order to achieve supplier minimums and/or container minimums on a particular day.

After the necessary minimums have been applied to the POs all planned orders that have met their lead time are released en mass to RMS for execution. The orders are communicated to RMS by using the RIB integration tool. The cron_release.sh perform the necessary steps to post orders to the appropriate RIB publication tables. The methods of interfacing purchase orders and transfers are dictated by separate system parameters which are stored in the database. In the event that the RIB is not used, AIP Oracle can extract purchase order and transfer data into text files.

The tsf_po_export script is used to accomplish this. However, it is important to note that RMS supports RIB-based purchase order and transfer subscription only.

In addition to regular data load, order load, data export, order export, order scale and order release processes, there is a process to remove closed orders in order to keep database table space available. The Order Purge process is used to purge orders that have been closed for more than their configured purge age (as defined in order_purge_period table). This process should be executed daily considering the volume of orders loaded and processed every day. It however is a non-critical process.

Post AIP Batch

The RIB publication of the AIP purchase orders and transfers to RMS completes the batch processes related to AIP.

Environment Variables

Before any batch scripts are run, the basic environment information must be globally defined for use in all the scripts. Environment variables and implementation parameters need to be set up for both the RPAS and the AIP Java/Oracle platforms.

- For information on the environment variables, refer to the, System Configuration chapter in the *Oracle Retail Advanced Inventory Planning Implementation Guide*.
- For information on the implementation parameters, refer to the AIP RPAS Configurations chapter in the *Oracle Retail Advanced Inventory Planning Implementation Guide*.

11

AIP RPAS Intra-day Batch Processing

The complete AIP Intra-day batch processing is comprised of the following:

- AIP RPAS Intra-day batch processing
- AIP Oracle Intra-day batch processing

Because the AIP solution resides on two platforms and needs to exchange information across both the platforms, the batches are executed on both the platforms. The intraday_batch.sh script is provided with the AIP installation, and it is used to run the entire RPAS Intra-day batch process from a UNIX scheduler. The scripts called by intraday_batch.sh may have multi-threaded process calls.

For more details on AIP Oracle Intra-day batch processing, refer to Chapter 6, "AIP Java/Oracle Batch Process Flow" and Chapter 7, "AIP Java/Oracle Daily Batch Process Details".

The AIP RPAS Intra-day Batch Control Script (intraday_batch.sh)

The AIP RPAS Intra-day batch script, intraday_batch.sh, accepts a number of arguments that allow more control over what portion of the batch scripts are run. Step names have been defined, which may also be passed into the script as arguments that define exactly which steps and corresponding scripts should be run.

Usage

Table 11–1 provides descriptions of the intraday_batch.sh arguments.

Table 11–1 Arguments for intraday_batch.sh

Argument	Description
-w	Wave number. This flag is required for all Intra-day batch run.
-s	Indicates that a starting step is defined. The step at which the batch should start must follow this flag. This flag can be used with all the other flags; however, only one batch step can follow the flag. This flag *must* be used along with the -e flag and its associated argument.
-e	Indicates that an ending step is defined. The step after which the batch should end must follow this flag. This flag can be used with all the other flags; however, only one batch step can follow the flag. This flag *must* be used along with the -s flag and its associated parameter.

Steps for intraday_batch.sh

The following is a list of valid steps that can be used with the -s and -e flags. These steps can also be passed as parameters to the intraday_batch.sh script, in the form of a list of steps (not flagged with -s and -e).

Execution Sequence of the AIP RPAS Intra-day Batch Scripts

Table 11–2 describes the steps in the order that intraday_batch.sh executes them.

Table 11–2 Execution Sequence of the AIP RPAS Intra-day Batch Scripts

Step Name	Description of Action
prep_intraday_inventory_data	This script calls _check_for_required_files.sh and process_intraday_inventory_data.sh. It first verifies the existence of all the required files as specified by the intraday_inventory_data.config file and then calls process_intraday_inventory_data.sh.
load_intraday_inventory_data	Preprocesses and loads all the Intra-day measures as specified by the intraday_inventory_data.config file.
run_intraday_replenishment_global	Runs Intra-day replenishment and reconciliation process at the global domain level.
prepare_intraday_local_domain_list	Creates a list of local domains to lock against during the subsequent, applicable Intra-day batch steps.
run_intraday_replenishment_local	Runs Intra-day replenishment and reconciliation at the local domain level.
export_intraday_replenishment_data	Creates the Intra-day replenishment plan extracts at the local domain level.
send_intraday_replenishment_data_to_om	Concatenates .dat files from an AIP domain's local domain output directories whose domains are specified in the id_local_domain.list into combined export files that are located in the global domains.
run_intraday_alerts	Updates the Intra-day alert indicator.
auto_build_intraday_workbooks_global	Builds all the Intra-day global workbooks that have been configured for automatic builds.
auto_build_intraday_workbooks_local	Builds all the Intra-day local workbooks that have been configured for automatic builds.

Example Script Calls

Table 11–3 lists example script calls and their actions.

Table 11–3 Intra-day Example Script Calls

Command	Action
intraday_batch.sh -w 7	Runs all the `intraday_batch.sh` steps against the wave 7
intraday_batch.sh -w 7 -s prep_intraday_inventory_data -e run_intraday_replenishment_local	Runs the `intraday_batch.sh` steps, starting with `prep_intraday_inventory_data`, and up to and including `run_intraday_replenishment_local`
intraday_batch.sh -w 7 run_intraday_alerts	Only runs the `intraday_batch.sh` step listed which is `run_intraday_alerts`
intraday_batch.sh -w 7 run_intraday_alerts auto_build_intraday_workbooks_global auto_build_intraday_workbooks_local	Results in the `intraday_batch.sh` running `run_intraday_alerts`, followed by `auto_build_intraday_workbooks_global` and `auto_build_intraday_workbooks_local batch`

12

AIP RPAS Intra-day Batch Scripts

Functionally the AIP RPAS Intra-day batch scripts are broadly classified in Table 12–1. Each of the scripts listed next to its descriptive step name exists to perform that batch step.

Table 12–1 AIP Intra-day Batch Scripts by Class

AIP Intra-day Batch Class	Step Description	Batch Script
Intra-day External Integration Inventory Data Processing and Load into AIP RPAS	Verify and Process Inventory Data from External System	check_process_intraday_inventory_data.sh
	Load Intra-day Inventory Data	load_intraday_inventory_data.sh
Run Intra-day Replenishment Process for Global Domain	Run Intra-day Replenishment for Global Domain	run_intraday_scrp_global.sh
Prepare Intra-day Local Domain List	Prepare Intra-day Local Domain List	prepare_intraday_localdomain_list.sh
Calculate and Export the Intra-day Replenishment Plans for Local Domains	Calculate Intra-day Replenishment Plans for Local Domains	run_intraday_scrp_local.sh
	Export Intra-day Replenishment Data for Local Domains	export_intraday_data_local.sh
Send Intra-day Replenishment Data from AIP RPAS to AIP Oracle	Send Intra-day Replenishment Data to OM	send_intraday_data_to_om.sh
Update Intra-day Alert Indicator	Calculate Intra-day Alert Indicator	update_intraday_alerts.sh
Build Intra-day Workbooks after Intra-day Batch Run	Auto Build Intra-day Global Workbooks	build_intraday_workbooks_global.sh
	Auto Build Intra-day Local Workbooks	build_intraday_workbooks_local.sh

Intra-day External Integration Inventory Data Processing and Load into AIP RPAS

These sections describe the steps needed for this process:

- Verify and Process Intra-day Inventory Data from External System
- Load Intra-day Inventory Data

Verify and Process Intra-day Inventory Data from External System

Step Name

prep_intraday_inventory_data

Script Call

check_process_intraday_inventory_data.sh

Parameters

None.

Functional Overview

The Intra-day data from RMS and/or non-RMS external system is processed as a set of flat files that follow an agreed AIP file format. AIP validates all of the files to ensure that the required data is present, and massages the files to produce the input required for subsequent steps of the AIP RPAS Intra-day batch.

Technical Details

The dynamic Intra-day measure data are listed in Table 12–2. They are also listed in the $AIPDOMAIN/interface/config/intraday/intraday_inventory_data.config. For AIP RPAS Intra-day batch to properly process external system data, all of the flat files must follow a particular format, which is explained in the *Oracle Retail Advanced Inventory Planning Implementation Guide*. The flat files must have a *.txt extension.

Table 12–2 Dynamic Measure Data

File Name	Description
idstrcurinvi.txt	Intra-day Loaded Store Current Inventory
idstriti.txt	Intra-day Store In Transits
idstrooi.txt	Intra-day Store On Orders
idwhcurinvi.txt	Intra-day Loaded Warehouse Current Inventory
idwhiti.txt	Intra-day Warehouse In Transits
idwhooi.txt	Intra-day Warehouse On Orders
idaiwi.txt	Intra-day WH Allocations in the Well
idtiwi.txt	Intra-day WH Transfers in the Well
iddayslsi.txt	Intra-day Loaded Daily Sales

Wave on Wave Processing

The following steps describe Wave on Wave processing,

1. Verify that all previous required files have been downloaded. Failure to download any of the required files results in an error and termination of the batch.
2. Prefix all the measure data listed in Table 12–2 except iddayslsi with the stocking point prefixes, using construct_ntier_measuredata.ksh:
   - idstrcurinvi.txt
   - idstriti.txt
   - idstrooi.txt

- idwhcurinvi.txt
- idwhiti.txt
- idwhooi.txt
- idaiwi.txt
- idtiwi.txt

3. Using interutil binary, convert from RMS SKU to AIP SKU all RMS-sourced inventory measure data except iddayslsi. See the earlier script and the *Oracle Retail Advanced Inventory Planning Implementation Guide* for details on this configuring.

   Note: The AIP configuration is to have all dynamic data listed in Table 12–2.

4. If the inventory tracking level is set to True, all like keys needs to be summed for all files except iddayslsi.

   Note: The reason daily sales is excluded from prefixing is that the prefixing is done inside `aipt_sr0_dayslsld_rms<x>.ksh`. The reason daily sales is excluded from interutil mapping is that it does not need RMS SKU/Order Mutliple -> SKPS mapping (it is at SKU). The reason it should be excluded from like-key summing is that this is unnecessary for those data feeds not run through interutil.

This Script Calls These Scripts:

- _check_for_required_files (defined in bsa_check_for_required_files.sh)
- process_intraday_inventory_data.sh

Appropriate Batch Run

Intra-day Wave.

Prerequisites

This step must not be initiated until successful completion of the following:

- All data files from Table 12–2 should be copied by the client's batch scheduler into the $AIPDOMAIN/interface/rms directory.
- Refer to the $AIPDOMAIN/interface/config/intraday/intraday_inventory_data.config for requirements.

Restart/Recovery

If the process_intraday_inventory_data.sh script failed, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Re-copy or re-FTP all required files listed in $AIPDOMAIN/interface/config/intraday/intraday_inventory_data.config into the $AIPDOMAIN/interface/rms directory.

   The new copies should overwrite the existing files in this directory.
4. Restart the batch.

Load Intra-day Inventory Data

Step Name

load_intraday_inventory_data

Script Call

run_ride.sh load_intraday_inventory_data.sh

Parameters

None.

Functional Overview

This script preprocesses and loads into the AIP RPAS domain all store and warehouse replenishment measure data that come from RMS or other external inventory system.

Technical Details

This script will loop over all measures in the intraday_inventory_data.config and move them to the AIP domain input directory and then load them into the domain. It will load all files as .ovr.

Wave on Wave Processing

Call loadmeasure.sh (a script wrapper for loadmeasure RPAS binary) on the measure data files corresponding to the Intra-day measure data received from the inventory system (example, RMS). The measures loaded are listed in the following configuration files:

```
$AIPDOMAIN/interface/config/intraday/intraday_inventory_data.config
```

Note: All measures are loaded as overlay (.ovr).

Note: For run_ride.sh:

This is a wrapper script for the RPAS ride utility which is introduced for Intra-day. The run_ride.sh, when being run against a global domain, will lock all domains; however, when the script is run against local domains, it will only lock those local domains maintained in a cached local domain list.

The usage for run_ride.sh is:

run_ride.sh [-local] script

If the flag -local is used, the script will lock only local domains listed in the cached list; otherwise, it will lock all domains.

This Script Calls This Script:

loadmeasure.sh

Appropriate Batch Run

Intra-day Wave.

Prerequisites

This step must not be initiated until successful completion of the following:

`check_process_intraday_inventory_data.sh`

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Ensure that the *.txt files are transferred through FTP by client scheduled process from RMS to the correct location.
4. Ensure all RMS file are present.
5. Ensure all files are formatted correctly.
6. Restart the batch.

Run Intra-day Replenishment Process for Global Domain

This section describes the steps needed for this process:

Step Name

run_intraday_replenishment_global

Script Call

run_ride.sh run_intraday_scrp_global.sh ${waveNumber}

Parameters

Wave Number

Functional Overview

The current wave measure which is a scalar is populated with the desired wave number before it can be used in the subsequent Intra-day batch steps.

Technical Details

This step initiates the current wave measure IdCurWavI based on the provided parameter.

Wave on Wave Processing

Populate the current wave measure to be used by subsequent Intra-day batch steps.

Appropriate Batch Run

Intra-day Wave.

Prerequisites

None.

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.

2. Correct any identified setup or environment issues.
3. Restart the batch.

Prepare Intra-day Local Domain List

This section describes the steps needed for this process:

Step Name

prepare_intraday_local_domain_list

Script Call

run_ride.sh prepare_intraday_localdomain_list.sh

Parameters

None.

Functional Overview

A list of local domains that need to be locked by run_ride.sh is prepared during this step for subsequent Intra-day batch calls which are run against local domains.

Technical Details

This step prepares a list of local domains to be used in subsequent, applicable Intra-day batch steps. This script first runs rule group id_preLocDomList, then exports measure IdLocDomListO to a temporary file, then generates the position list file and local domain list file.

Table 12–3 Rule Groups Used for Preparing Intra-day Local Domain List

Rule Group	Description
id_preLocDomList	Prepare a list of local domains to be used in subsequent Intra-day batch steps.

Appropriate Batch Run

Intra-day Wave.

Prerequisites

This step must not be initiated until successful completion of the following:

run_ride.sh run_intraday_scrp_global.sh ${waveNumber}

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Calculate and Export the Intra-day Replenishment Plans for Local Domains

These sections describe the steps needed for this process:

Calculate Intra-day Replenishment Plans for Local Domains

Step Name

run_intraday_replenishment_local

Script Call

run_ride.sh -local for_each_local_domain.sh -l $INTRADAY_LOCAL_DOMAIN_LIST -p run_intraday_scrp_local.sh [DOMAIN]

Parameters

Path to Local Domain

Functional Overview

Intra-day Replenishment and Reconciliation is part of the AIP batch run which generates an Intra-day Receipt Plan from Warehouses or Supplier to Stores across the length of the horizon. The final output from this Intra-day batch process is:

- Intra-day Constrained Receipt Plan for Stores across the Fixed Period.
- Unconstrained Intra-day Receipt Plan for Stores across the post Fixed Period up to the end of the horizon.

Technical Details

run_intraday_scrp_local.sh on all local domains. The local scripts internally invoke a set of rule groups according to the sequence described in Table 12–4. Each rule group is responsible for performing a specific step in the supply chain replenishment planning process.

Table 12–4 lists rule groups involved in the AIP local Intra-day replenishment and reconciliation batch process.

Table 12–4 Rule Groups for AIP Local Intra-day Batch Process

Rule Group	Description
Local Rule Groups	The following rule groups are called by run_intraday_scrp_local.sh script.
id_setup	Run all Intra-day setup steps for AIP.
id_preReplenish	Runs all Intra-day pre replenishment steps for AIP.
id_replenish	Runs Intra-day replenishment batch run and generates an unconstrained receipt plan.
id_preReconcile	Runs Intra-day pre reconciliation steps for AIP.
id_reconcile	Runs Intra-day reconciliation and generates an Intra-day constrained receipt plan.
id_pushToStore	Runs Intra-day stockless reconciliation at store destinations for stockless items and pushes the excess quantity to the stores.
id_replPstFxd	Runs Intra-day replenishment batch run post fixed period and generates an Intra-day unconstrained receipt plan for post fixed period.

Table 12–4 (Cont.) Rule Groups for AIP Local Intra-day Batch Process

Rule Group	Description
id_post	Runs Intra-day post replenishment process.

Wave on Wave Processing

The run_intraday_scrp_local.sh will call rule groups according to the previous orders against the local domains produced by the previous step.

Appropriate Batch Run

Intra-day Wave.

Prerequisites

This step must not be initiated until successful completion of the following:

- run_ride.sh run_intraday_scrp_global.sh ${waveNumber}
- run_ride.sh prepare_intraday_localdomain_list.sh

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Export Intra-day Replenishment Data for Local Domains

Step Name

export_intraday_replenishment_data

Script Call

run_ride.sh -local for_each_local_domain.sh -l $INTRADAY_LOCAL_DOMAIN_LIST -p export_intraday_data_local.sh [DOMAIN]

Parameters

Path to Local Domain

Functional Overview

The replenishment plan's planned orders from suppliers and transfers from warehouses must be exported from populated measures into files suitable for interfacing with AIP Oracle. In this batch step, the exports are performed independently in each local domain, resulting in each having its own copy of the export files placed in the local domain output directory, as shown in Table 12–5.

Table 12–5 Export Files

File Name	Description
strsplrord.dat	Supplier to store orders for release today through the planning horizon.
strwhord.dat	Warehouse to store transfers for release today.

Technical Details

This step's command line script call is a composite of two scripts, for_each_local_domain.sh and export_intraday_data_local.sh, which together cause the exporting of order and transfer data to be performed in parallel across all local domains. At the local domain level, the process is driven by the aipcmd binary, which processes the XML resource command files to produce the parenthesized output files:

- exportPlanStrSplrIdOrd.xml (strsplrord.dat)
- exportPlanStrWhIdOrd.xml (strwhord.dat)

These XML resource files all make use of the ExportPlan command to perform the specialized data exports.

Wave on Wave Processing

Script for_each_local_domain.sh performs the following steps:

1. Determine the list of local domains under $AIPDOMAIN.
2. For each local domain, call in parallel the script export_intraday_data_local.sh and pass it the local domain path. That path is substituted automatically for the DOMAIN] token by for_each_local_domain.sh.

Appropriate Batch Run

Intra-day Wave.

Prerequisites

This step must not be initiated until successful completion of the following:

run_ride.sh -local for_each_local_domain.sh -l $INTRADAY_LOCAL_DOMAIN_LIST -p run_intraday_scrp_local.sh [DOMAIN]

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Send Intra-day Replenishment Data from AIP RPAS to AIP Oracle

This section describes the steps needed for this process:

Step Name

send_intraday_replenishment_data_to_om

Script Call

send_intraday_data_to_om.sh

Parameters

None.

Functional Overview

Planned orders and transfers, exported separately per local domain in the Export Replenishment Data step, are combined, tarred, and compressed for interfacing with AIP Online.

Technical Details

In this batch step, the filenames listed in the global domain's $AIPDOMAIN/interface/config/intraday/intraday_scrp_export_to_om.config are located in each local domain's output directory and concatenated into like-named files in the global domain's $AIPDOMAIN/interface/export directory.

Wave on Wave Processing

1. Verify the existence and writeability of the global domain's interface/export directory, the destination of the concatenated export files.
2. For each of the files listed in the global domain's $AIPDOMAIN/interface/config/intraday/intraday_scrp_export_to_om.config file, delete any existing instance in the global domain's $AIPDOMAIN/interface/export directory, then iterate over the list of local domains, concatenating each instance of the file found in the local domain's output directory into the global interface file of the same name in global domain's $AIPDOMAIN/interface/export directory.
3. Compress each global domain export file, package them all into a single tar file, then compress the tar file into a single file:

   $AIPDOMAIN/interface/export/srp.tar.Z.

Appropriate Batch Run

Intra-day Wave.

Prerequisites

This step must not be initiated until successful completion of the following:

run_ride.sh -local for_each_local_domain.sh -l $INTRADAY_LOCAL_DOMAIN_LIST -p export_intraday_data_local.sh [DOMAIN]

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Update Intra-day Alert Indicator

This section describes the steps needed for this process:

Step Name

run_intraday_alerts

Script Call

for_each_local_domain.sh -p update_intraday_alerts.sh [DOMAIN]

Parameters

Path to Local Domain

Functional Overview

This step calculates the Intra-day Out of Stock Indicator which then be used by the next overnight batch for calculating the SRP Alerts.

Technical Details

This step updates the Intra-day Out of Stock Indicator after the Intra-day batch jobs have finished.

Wave on Wave Processing

Execute id_srpAlerts rule group to update the Intra-day Out of Stock Indicator (IdOosI) for later overnight batch use.

Appropriate Batch Run

Intra-day Wave.

Prerequisites

This step must not be initiated until successful completion of the following:

run_ride.sh -local for_each_local_domain.sh -l $INTRADAY_LOCAL_DOMAIN_LIST -p run_intraday_scrp_local.sh [DOMAIN]

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Build Intra-day Workbooks after Intra-day Batch Run

These sections describe the steps needed for this process:

- Auto Build Intra-day Global Workbooks
- Auto Build Intra-day Local Workbooks

Auto Build Intra-day Global Workbooks

This section describes the steps needed for this process:

Step Name

auto_build_intraday_workbooks_global

Script Call

run_ride.sh build_intraday_workbooks_global.sh ${waveNumber}

Parameters

Wave Number

Functional Overview

Many of the Intra-day workbooks can be configured by the system administrator to be automatically built during the day as part of the AIP RPAS Intra-day batch run. This allows you to enter the worksheets directly without going through the workbook creation wizard. All Intra-day workbooks configured for commit at global domain level are committed in this step. This script may also be run from the command line ad hoc throughout the day by administrators or users.

> **Note:** Only a workbook that is in the Intra-day category can be built during the Intra-day batch run. None of the workbooks other than Intra-day workbooks will change during the day.

Technical Details

This step uses the RPAS utility wbbatch to automatically build Intra-day workbooks. In order to successfully auto-build workbooks, they must be configured through the RPAS client. Refer to the *Oracle Retail Predictive Application Server Administration Guide* for information on automatically building workbooks.

Wave on Wave Processing

Call wbbatch to build all workbooks on the $AIPDOMAIN global domain build queue.

Appropriate Batch Run

Intra-day Wave.

Prerequisites

This step must not be initiated until the following steps (scripts) are successfully completed.

run_ride.sh -local for_each_local_domain.sh -l $INTRADAY_LOCAL_DOMAIN_LIST -p run_intraday_scrp_local.sh [DOMAIN]

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.

Auto Build Intra-day Local Workbooks

This section describes the steps needed for this process:

Step Name

auto_build_intraday_workbooks_local

Script Call

run_ride.sh -local for_each_local_domain.sh -l $INTRADAY_LOCAL_DOMAIN_LIST -p build_intraday_workbooks_local.sh [DOMAIN] ${waveNumber}

Parameters

Path to Local Domain and Wave Number

Functional Overview

Many of the Intra-day workbooks can be configured by the system administrator to be automatically built during the day as part of the AIP RPAS Intra-day batch run. This allows you to enter the worksheets directly without going through the workbook creation wizard. All Intra-day workbooks configured for commit at local domain level are committed in this step. This script may also be run from the command line ad hoc throughout the day by administrators or users.

Technical Details

This step uses the RPAS utility wbbatch to automatically build Intra-day workbooks. In order to successfully auto-build workbooks, they must be configured through the RPAS client. Refer to the *Oracle Retail Predictive Application Server Administration Guide* for information on automatically building workbooks.

Note: Only workbook that is in the Intra-day category can be built during the Intra-day batch run. None of the workbooks other than Intra-day workbook will change during the day.

Wave on Wave Processing

For each local domain, call wbbatch to build all Intra-day workbooks on each local domain build queue.

Appropriate Batch Run

Intra-day Wave.

Prerequisites

This step must not be initiated until the following steps (scripts) are successfully completed.

run_ride.sh -local for_each_local_domain.sh -l $INTRADAY_LOCAL_DOMAIN_LIST -p run_intraday_scrp_local.sh [DOMAIN]

Restart/Recovery

If this script fails, perform the following steps:

1. Examine the log files to determine the cause of the failure.
2. Correct any identified setup or environment issues.
3. Restart the batch.