

Oracle® Retail Merchandising

Batch Schedule

Release 12.0.8

July 2008

Copyright © 2008, Oracle. All rights reserved.

Primary Author: Rich Olson

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

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

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

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

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

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

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

Value-Added Reseller (VAR) Language

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

Contents

Preface	vii
Audience	vii
Related Documents.....	vii
Customer Support.....	viii
Conventions.....	viii
1 Introduction to Merchandising Batch Processing	1
Batch Processing.....	1
Types of Batch Programs	1
Batch Window	2
Batch Schedule and Phases.....	2
Integrated Merchandising Batch Schedule.....	3
Program List	3
Batch Schedule Diagram	5
RMS, ReIM, RTM Section	5
ReSA Section.....	6
RPM Section.....	6
Notations in the Batch Schedule Diagram.....	7
prepost Program	8
Modifications to the Batch Schedule	9
2 Program List.....	11
3 Batch Schedule Diagram	17
4 Interface Diagrams for RMS and RPAS	19
RMS Pre/Post Extract Diagrams	20
RMS Foundation Data Extract Diagrams	21
RMS Fact Data Extract Diagrams.....	23
RPAS-RMS Fact Load Diagram	24
5 Interface Diagrams for RMS and RDW.....	25
6 Interface Diagram for RPM and RDW.....	37
7 Interface Diagram for ReIM and RDW	39

Preface

This batch schedule document details the integrated cyclical processing schedules for the Oracle Retail Merchandising applications:

- Oracle Retail Merchandising System (RMS)
- Oracle Retail Invoice Matching (ReIM)
- Oracle Retail Price Management (RPM)
- Oracle Retail Sales Audit (ReSA)
- Oracle Retail Trade Management (RTM)
- Oracle Retail Allocation

Note: Although Oracle Retail Allocation is a Merchandising application, it is not represented in this batch schedule because it does not have any batch programs to run. All Allocation processing is online processing.

This guide describes the periodic and ad hoc phases of batch processing, as well as pre- and post-processing dependencies.

Audience

The audiences for this guide are as follows:

- Systems analysts and system operations personnel who need information about Merchandising processes, internally or in relation to systems across the enterprise
- Integrators and implementation staff who have the overall responsibility for implementing the Merchandising applications in their enterprise

Related Documents

For more information, see the following documents for the Oracle Retail Merchandising products:

- Oracle Retail Merchandising Implementation Guide
- Oracle Retail Merchandising System Operations Guide
- Oracle Retail Price Management Operations Guide
- Oracle Retail Invoice Matching Operations Guide
- Oracle Retail Data Warehouse Operations Guide
- Oracle Retail Predictive Application Server documentation
- Oracle Retail Demand Forecasting documentation

Customer Support

<https://metalink.oracle.com>

When contacting Customer Support, please provide the following:

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

Conventions

Navigate: This is a navigate statement. It tells you how to get to the start of the procedure and ends with a screen shot of the starting point and the statement “the Window Name window opens.”

Note: This is a note. It is used to call out information that is important, but not necessarily part of the procedure.

This is a code sample
It is used to display examples of code

A hyperlink appears like this.

Introduction to Merchandising Batch Processing

This chapter is a brief introduction to Oracle Retail batch processing. It defines basic terms and concepts, describes batch processing phases, and explains how to interpret the batch schedule diagram and program list.

Batch Processing

Batch processing is the execution of a group of batch programs (jobs). The results are returned without user intervention. Batch programs are commonly used for the following reasons:

- To process large volumes of transaction data
- To interface with external systems
- To perform internal maintenance

Batch programs can process very large quantities of data quickly and efficiently. Batch programs can perform some updates that could be performed through online transactions, but much more quickly and with less impact on system performance. Batch processing is usually scheduled for times when systems are idle or least busy.

Batch programs can be run automatically using batch scheduler software. The batch scheduler allows batch jobs to be set up in a specific order, with restrictions attached to any program as needed. If an error occurs with a batch program, an administrator must correct the error and manually rerun the batch program that failed.

Types of Batch Programs

Oracle Retail batch programs are of several types:

- Upload programs bring data from external systems into the Oracle Retail database. For example, the `posupld` program uploads daily transactions that occur at the point of sale (POS) for processing by the Oracle Retail Management System (RMS).
- Download programs extract data from RMS and format it so it can be used by external systems. For example, the `posdnld` program extracts new and changed information about an item/location for downloading to the point of sale.
- System maintenance programs perform tasks such as updating the system date. For example, the `dtesys` program increments the system date at the end of each batch cycle.
- Functional maintenance programs process data specific to a functional area. For example, the `storeadd` program updates a number of tables to create entries for a new store.

Batch Window

Because of the impact on production systems, it is not always possible to run batch programs during business hours; however, there is a window of opportunity during each day or night when online systems are not being used. This time frame is the *batch window*. For example, a retailer with stores throughout the continental U.S. might require its online systems to be available from 8 AM Eastern Standard Time, when its East Coast offices open, until 9 PM Pacific Standard Time, when its West Coast stores close. This allows an eight-hour batch window for processing all batch jobs.

Batch Schedule and Phases

Order is critical when running batch programs. Some tasks need to be performed before others. A batch schedule ensures that every time batch processing is performed, the correct tasks are performed in the proper order.

The batch schedule is a diagram that represents all batch programs and how they are sequenced. For each individual user, the schedule is a suggested starting point for the installation. Some programs are specific to products that may not be installed, so these programs may not be used at all.

The total batch schedule is divided into phases. Each phase must be completed before the next phase can begin. Within a phase, there may also be programs that depend on the completion of another program within that phase, so programs within each phase may need to be run in a particular order.

Merchandising Batch Schedule

The integrated Merchandising batch schedule combines the batch schedules of all Merchandising applications into a single schedule diagram. The diagram (later in this document) shows the batch dependencies among the Merchandising applications.

The integrated Merchandising batch schedule combines the batch modules for the following applications:

- Oracle Retail Merchandising System (RMS)
- Oracle Retail Trade Management (RTM)
- Oracle Retail Sales Audit (ReSA)
- Oracle Retail Invoice Matching (ReIM)
- Oracle Retail Price Management (RPM)

Note: Although Oracle Retail Allocation is a Merchandising application, it is not represented in this batch schedule because it does not have any batch programs to run. All Allocation processing is online processing.

Program List

The columns of the program list provide details about each batch program, as follows:

Column	Description
Program name	Name of the program or script
Functional area	Functional area of the application for which the batch program is run
Threaded	Whether the program is threaded (Y/N)
Driver	Program driver
Phase	Phase during which the program is run (see the batch schedule diagram)
Pre-dependency	Programs that must be completed before the program can be run
Post-dependency	Programs that must be run after the program completes successfully
Timing	How often the program is run (for example, daily, weekly, monthly, ad hoc)
Restart/Recovery	Whether the program uses restart/recovery (R=Yes, N=No)
Run Parameters for Program	Command syntax to run the program

For example, the following shows the information in the program list about an RMS phase 3 program named dealday:

Program Name	dealday
Functional Area	Deals
Threaded	Y
Driver	Location
Phase	3
Pre-dependency	dealinc, dealfinc, prepost dealday pre
Post-dependency	prepost dealday post, salmnth
Timing	Monthly
Restart/Recovery	R
Usage	dealday userid/passwd

The program list is grouped in the following order:

- RMS, RTM, and ReSA programs
- RPM programs
- ReIM programs
- RMS extracts for Retail Predictive Application Server (RPAS)
- RMS extracts for Retail Data Warehouse (RDW)

The extracts for RPAS and RDW are programs that are part of the RMS application.

Batch Schedule Diagram

The batch schedule diagram illustrates the program list pre- and post-dependency details. The layout and notations of the diagram also illustrate required sequences and other processing details. Executing the Merchandising batch processing in the manner diagrammed ensures that all critical dependencies are met.

For ease of setting up a schedule at client site, and also based on logical application dependencies, the diagram is divided into three main sections:

- RMS, RTM, ReIM
- ReSA
- RPM

Later chapters of this document show data flow diagrams for other batch processes:

- Chapter 4 shows the Retail Extract, Transform, and Load (RETL) data flows for the extracts from RMS to RPAS.
- Chapter 5 shows the RETL dimension and fact data flows for the extracts from RMS to Retail Data Warehouse (RDW).
- Chapter 6 shows the RETL data flow for the Promotion dimension extract from RPM to RDW.
- Chapter 7 shows the RETL data flow for the Supplier Invoice Cost dimension extract from ReIM to RDW.

RMS, ReIM, RTM Section

The first section diagrams the RMS, ReIM, and RTM programs and their dependencies. This section is further divided into phases 0 through 7, ad hoc, and date set batch.

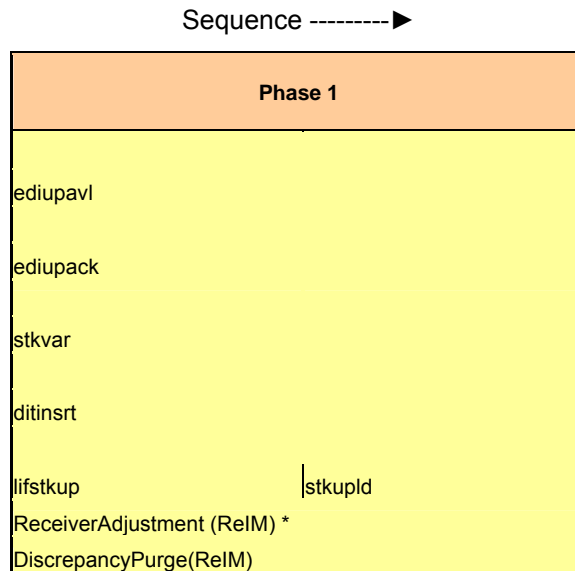
Each phase must be completed before the next phase can begin. Also, a phase may contain programs that depend on other programs within the phase. Programs within each phase may need to run in a particular sequence.

The following are brief descriptions of the Merchandising batch processing phases. Depending on your implementation, some programs and phases may not apply.

Phase	Description
Phase 0	The first phase performs essential table maintenance including: <ul style="list-style-type: none"> ▪ Daily purges ▪ Updates to currency exchange rates ▪ Updates to value-added tax (VAT) data
Phase 1	This phase prepares the tables for interfacing with external systems in Phase 2. Among other programs, the stock variance (stkvar) batch program is run to update stock counts.
Phase 2	During this phase, information is uploaded from external interfaces, including point of sale (POS) data (posupld batch program).
Phase 3	In this phase, the main RMS processing programs are run for purchasing, ordering, stock ledger, deals, and replenishment.
Phase 4	This phase pushes data to external sources. Changed system information is rebuilt. Open to buy (OTB) data is updated. Information is sent to the forecasting system.

Phase	Description
Phase 5	This phase consists of ReIM process upload programs.
Phase 6	This phase consists of ReIM process roll-up programs.
Phase 7	This phase consists of ReIM process download programs.
Ad Hoc	Ad hoc batch programs can be run at any time. The ad hoc programs have no phase dependencies.
Date Set	The Date Set phase increments the system date and updates other calendar dates. Note: The date set phase should be the very last phase to run. Even the ad hoc programs should be run before the date set program.

Read the batch schedule diagram from left to right. In the following example, any of the programs (ediupavl, ediupack, stkvar, ditinsrt, lifstkup, ReceiverAdjustment, DiscrepancyPurge) can start at the same time; however, the stkupld program cannot start until the lifstkup program is successfully completed.



ReSA Section

This section diagrams the ReSA programs and their dependencies.

RPM Section

This section diagrams the RPM programs and their dependencies.

Notations in the Batch Schedule Diagram

Pipes

Pipes are vertical bars (|) that represent the dependencies within a phase. Reading left to right, a pipe indicates that one or more programs to the right depend upon completion of one or more programs to the left.

In the following example, the stkupld module depends on the lifstkup module; that is, the stkupld module can be run only after successful completion of the lifstkup module.

lifstkup	stkupld
----------	---------

In the following example, both of the modules cntnrddb and reqext are dependent on ociroq. Neither cntnrddb nor reqext can be run until the ociroq module has completed successfully.

ociroq	cntnrddb reqext
--------	--------------------

In the following example, the ibcalc module is dependent on both ibexpl and cntprss. The ibcalc module cannot be run until both ibexpl and cntprss have completed successfully.

ibexpl cntprss	ibcalc
-------------------	--------

Abbreviations

In the diagram, abbreviations in parentheses that follow program names have the following meanings:

Abbreviation	Meaning
(perl)	The module is a Perl script.
(FIF)	The module is related to the Financials application.
(sqlldr)	There is a sqlloader process to load / ftp the output files.
(rebuild all)	There is a rebuild process inside the application.
(IM)	The module is related to Invoice Matching but owned by RMS.
(RMS)	The module belongs to RMS.
(RMS)	(Bold type) The RMS module is executed externally to that phase.
(ReSA)	The module belongs to ReSA.
(ReSA)	(Bold type) The ReSA module is executed externally to that phase.
(ReIM)	The module belongs to ReIM.
(RTM)	The module belongs to RTM.
(Weekly)	The module is executed weekly.
(Monthly)	The module is executed monthly.
(Forms Auditing)	This is an online forms auditing process related to ReSA.

Footnotes

Footnote symbols (*, **, †, ‡) refer to footnotes that appear below that phase or section of the diagram.

prepost Program

The prepost program facilitates multi-threading by allowing general system administration functions (such as table deletions or mass updates) to be completed after all threads of a particular program have been processed. The prepost program must be run before, after, or both before and after, programs that require specific processing to run or complete successfully.

In the batch schedule diagram, the prepost program is indicated by “pre” and “post” entries, as in the following examples.

In the following example, pre-processing is required before running the ociroq program.

pre	ociroq
------------	---------------

In the following example, pre-processing is required before running the stkupd program. Also, post-processing is required after successful completion of the stkupd program.

pre	stkupd	post
------------	---------------	-------------

In the following example, post-processing is required after successful completion of the sccext program.

sccext	post
---------------	-------------

Modifications to the Batch Schedule

The integrated Merchandising batch schedule shows the dependencies for all the programs that *could* be run by a retailer. Based on many factors, there will always be some programs that a retailer does not run. Determining which programs, or groups of programs, are not required is a job that should be performed at implementation time.

One major factor involves the applications that the retailer has purchased and wants to install:

- For example, a retailer may have purchased RMS, but not ReIM; in this case, the ReIM programs would not be run.
- Another example is that a retailer may not want to use some functionality within an application. Perhaps a retailer purchased RMS but did not purchase the RDW application. In this case, the retailer may not want to run the programs that extract RMS data to be used later by the RDW application.

These major configuration choices also affect whether some programs are used:

- Whether the Retail Integration Bus (RIB) is used
For more information about configuring the RIB for Merchandising applications, see “Configuring RPM without the RIB” in the “Backend System Administration and Configuration” chapter of the Retail Price Management Operations Guide.
- Whether full-featured or simplified Retail Price Management (RPM) is used
For more information about configuring simplified RPM, see the “Backend System Administration and Configuration” chapter in the Retail Price Management Operations Guide.
- Whether full-featured or simplified RTM is used
For more information about configuring simplified RTM, see the “Oracle Retail Trade Management Batch” chapter in Volume 1 of the Retail Merchandising System Operations Guide.

RMS,RTM,ReSA Program Dependency and Scheduling Details

Program Name	Functional Area	Threaded	Driver	Phase	Pre-dependency	Post-dependency	Timing	Uses Restart/Recovery	Run Parameters for Programs
audtprg	Audit	N	N/A	ad hoc	N/A	N/A	daily	N	audtprg userid/passwd
audtsys	Audit	N	N/A	ad hoc	N/A	N/A	daily	N	audtsys userid/passwd
batch_orpos_extract.ksh	Point of Sale Interface	Y	Store	4	If RPM pricing info is reqd then run after extraction script	prepost poscdnid post	daily	N	batch_orpos_extract.ksh [-p <no. of threads>] userid/passwd [DIR - location where extracts are to be generated]
ccprg	Costing	N	N/A	ad hoc	RPMtoORFOSPublishExport.sh'	prepost poscdnid post	monthly	N	ccprg userid/passwd
ccdnid	Trade Management	Y	Broker	2	N/A	N/A	daily	R	ccdnid userid/passwd broker file_name
cmpprg	Pricing	N	N/A	ad hoc	N/A	N/A	daily	N	cmpprg userid/passwd
cmpupld	Pricing	N	N/A	ad hoc	N/A	All RPM batch modules	ad hoc	R	cmpupld userid/passwd input_file reject_file
cntrmain	Contracting	N	N/A	0	All Replenishment modules	cntrrordb post	daily	R	cntrmain userid/passwd
cntrrordb	Contracting	Y	Contract	3	rpladl	prepost cntrrordb post	daily	R	cntrrordb userid/passwd
cntrprss	Contracting	Y	Dept	3	rplast	cntrprss userid/passwd	daily	R	cntrprss userid/passwd
costcalc	Deals	Y	Supplier	2	precostcalc	prepost costcalc post	daily	R	costcalc userid/passwd supplier (May use the batch_costcalc.ksh for launching this program as it is created based o
cremhierdy	Reclassification	N	N/A	4	N/A	recsldy	daily	R	performance considerations)
deallact	Deals	Y	Deal Id	3	prepost deallact_pre	N/A	daily	R	deallact userid/passwd
dealcis	Deals	N	N/A	3	prepost deallact_pre	N/A	daily	R	dealcis userid/passwd
dealday	Deals	Y	Location	3	prepost deallact_pre	salnmth	monthly	R	dealday userid/passwd
dealex	Deals	Y	Deal Id	3	precostcalc	dealcis	daily	N	dealex userid/passwd
deallct	Deals	Y	Deal Id	3	prepost deallct_pre	salnmth	daily	R	deallct userid/passwd [Y/N - EOM processing ind]
deallnc	Deals	Y	Deal Id	3	deallact	deallct	weekly/ad hoc	R	deallnc userid/passwd
dealinc	Deals	Y	Deal Id	3	deallact	salnmth (if monthly)	monthly	R	dealinc userid/passwd [Y/N -EOM processing ind]
dealprg	Deals	N	N/A	ad hoc	prepost dealinc pre	N/A	monthly	R	dealprg userid/passwd
dealupld	Deals	Y	File-based	0	(This program is the first one in Deals batch	N/A	daily	R	dealupld userid/passwd input_file reject_file
dfrtbd	Item Maintenance	Y	Dept	3	information is uploaded into Oracle Retail)	(SQL*Load the output file)	daily	R	dfrtbd userid/passwd outfile
disotbaply	OTB	Y	OTB	4	ordscnt	N/A	daily	R	disotbaply userid/passwd
distrocpub	Pricing/Transfers/Allocation Publish	Y	Store	3	PriceEventExecutionBatch(RPM)	N/A	daily	R	distrocpub userid/passwd
dtinsrt	Deals	N	N/A	1	N/A	costcalc	daily	R	dtinsrt userid/passwd (P or S) (supplier/partner).
dyprg	Maintenance	N	N/A	0	N/A	(All other batch programs)	daily	N	Partner or Supplier.
docclose	Receiving	N	N/A	ad hoc	N/A	N/A	daily	R	appropriate calling script and passed into program. Note: (May use the batch_dtinsrt.ksh for launching this program as it
dtesys	Calendar	N	N/A	date_set	(This program should run at the end of the batch cycle)	prepost dtesys post	daily	N	is created based on performance considerations)
dummychn	Receiving	N	N/A	ad hoc	N/A	N/A	daily	N	dyprg userid/passwd
eddiadd	Maintenance	N	N/A	ad hoc	N/A	N/A	ad hoc	N	docclose userid/passwd
edidcon	Contracting	N	N/A	ad hoc	N/A	N/A	ad hoc	N	edidcon userid/passwd [indate-YYYYMMDD format]
edidlinv	Invoice Matching	Y	Location	4	N/A	N/A	daily	R	edididd userid/passwd eddiadd_output eddiadd_catalog
edidord	Ordering	N	N/A	4	ordrev	N/A	ad hoc	R	edidicon userid/passwd edidcon_outfil
edidprd	EDI Interface - Sales and Inventory	N	N/A	4	(and after replenishment batch)	prepost edidprd post	daily	R	edidlinv userid/passwd output_filename
ediprg	EDI Interface - Purge	N	N/A	ad hoc	prepost edidprd pre	prepost edidprd post	monthly	R	edidford userid/passwd filename
ediupadd	Maintenance	N	File-based	2	(Towards the end of the batch cycle	N/A	daily	N	edidiprd userid/passwd filename
ediupack	EDI Interface - ordering	N	N/A	1	N/A	N/A	ad hoc	R	ediprg userid/passwd
ediupavl	EDI Interface - Contracts	N	File-based	1	N/A	N/A	ad hoc	R	ediupadd userid/passwd input_file reject_file
ediupcat	EDI Interface - Suppliers	N	File-based	ad hoc	N/A	N/A	daily	R	ediupack userid/passwd data_file reject_file
fcstprg	Forecasting	Y	Domain Id	ad hoc	prepost fcstprg pre	prepost fcstprg post	daily	N	ediupavl userid/passwd input_file reject_file
fcstrbld	Forecasting	Y	Domain Id	3	N/A	prepost fcstrbld post	weekly	R	ediupcat userid/passwd edi_data_file error_fil
fcstrbld_sbc	Forecasting	Y	Domain Id	3	prepost fcstrbld post	N/A	weekly	R	fcstprg userid/passwd domain
fflgldn1	Financial Interface	Y	Dept	3	salstage	prepost fflgldn1 post	daily	R	fcstrbld userid/passwd
fflgldn2	Financial Interface	Y	Dept	3	salstage	salapnd	daily	R	fcstrbld_sbc userid/passwd
fflgldn3	Financial Interface	Y	Store/Wh	3	salnmth	salapnd	monthly	R	fflgldn1 userid/passwd
frmedrid	Planning System Interface	N	N/A	ad hoc	N/A	N/A	ad hoc	R	fflgldn2 userid/passwd
goupld	Misc Interface - Taxgeocode	N	N/A	ad hoc	N/A	N/A	ad hoc	R	fflgldn3 userid/passwd
genpreiss	Ordering	Y	Supplier	ad hoc	N/A	N/A	ad hoc	R	frmedrid userid/passwd
gradupld	Forecasting	N	File-based	ad hoc	N/A	N/A	ad hoc	R	goupld userid/passwd
hstbld	Sales	Y	Location	3	posupld	prepost hstbld post	weekly	R	genpreiss userid/passwd
hstbld_diff	Sales	N	N/A	ad hoc	prepost hstbld pre (for rebuild all)	prepost hstbld post	ad hoc	N	gradupld userid/passwd input_file rej_fil
hstbldmth	Sales	Y	Dept	3	hstbld	prepost hstbldmth post	monthly	R	hstbld userid/passwd level(weekly/rebuild)
hstbldmth_diff	Sales	N	N/A	ad hoc	posupld	prepost hstbldmth post	ad hoc	N	hstbld_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	(Run SQL*Loader using the control file hstbldmth.ctl to load data from the output file written by	ad hoc	R	hstbldmth userid/passwd level(monthly/rebuild)
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff	Sales	N	N/A	ad hoc	N/A	hstbldmth.ctl to load data from the output file written by HSTWKUPD.PC for non-existent records on	ad hoc	R	hstbldmth_diff userid/passwd
hstbldmth_diff</									

invaprg	Inventory Adjustments	N	N/A	ad hoc	N/A	N/A	monthly	N	invaprg userid/passwc
invclshp	Invoice Matching	N	N/A	2	N/A	N/A	daily	N	invclshp userid/passwc
invprg	Invoice Matching	N	N/A	ad hoc	ordbrg	N/A	monthly	R	invprg userid/passwc
lclcnld	Letter of Credit	N	N/A	4	N/A	lcm700 (perl script)	daily	R	lclcnld userid/passwd output_file
lclrbld	Maintenance - Location	N	N/A	ad hoc	storeadd	N/A	monthly	R	lclrbld userid/passwd
lclmndld	Letter of Credit	N	N/A	4	N/A	lcm707 (perl script)	daily	R	lclmndld userid/passwd output_file
lclup798	Letter of Credit	N	N/A	2	lcm798 (perl script)	N/A	daily	R	lclup798 userid/passwd input_file rej_fld
lclupld	Letter of Credit	N	N/A	2	lcm730 (perl script)	N/A	daily	R	lclupld userid/passwd input_file rej_file
lflstkup	Stock Ledger	N	File-based	1	inv_bal_upload.sh (warehouse mgmt program)	stakupd	daily	N	lflstkup userid/passwd input_file output_file
lflkstore	Maintenance - Location	Y	Dept	ad hoc	storeadd	prepost lkstore pos	daily	R	lflkstore userid/passwc
mrt	Mass Return Transfers	Y	Warehouse	2	N/A	mrttrv	daily	R	mrt userid/passwd
mrtprg	Mass Return Transfers	Y	Warehouse	ad hoc	N/A	mrtupd	ad hoc	R	mrtprg userid/passwd
mrttrv	Mass Return Transfers	Y	Warehouse	2	mrt	mrtupd	daily	R	mrttrv userid/passwd
mrtupd	Mass Return Transfers	Y	Warehouse	2	mrttrv	N/A	daily	R	mrtupd userid/passwd
nwppurge	Stock Ledger	N	N/A	ad hoc	N/A	N/A	ad hoc	N	nwppurge userid/passwd
nwpyearend	Stock Count	Y	Location	4	run on last day of yea	N/A	yearly	R	nwpyearend userid/passwc
ocroq	Replenishment	N	N/A	3	repladj	N/A	daily	R	ocroq userid/passwd
onictext	Planning System Interface	Y	Transfer	4	onordext	onordndld	weekly	R	onictext userid/passwd datefil
onordndld	Planning System Interface	Y	Store/Wh	4	onictext	N/A	daily	R	onordndld userid/passwc
onordext	Planning System Interface	Y	Order	4	prepost onordext pr	onictext	daily	R	onordext userid/passwd datefil
ordautcl	Ordering	N	N/A	ad hoc	N/A	N/A	daily	N	ordautcl userid/passwc
orddsct	Deals	Y	Supplier	4	disinst	discothapply	daily	R	orddsct userid/passwd
ordbrg	Ordering	N	N/A	ad hoc	reclsdly	dealcsl	monthly	R	ordbrg userid/passwc
ordrev	Ordering	N	N/A	4	orddsct	edidford	daily	R	ordrev userid/passwc
ordupd	Ordering	N	N/A	4	scext	otbdld	daily	N	ordupd userid/passwd
otbdld	OTB	N	N/A	4	(After RPM pricing change extraction batch)	otbdld	daily	R	otbdld userid/passwd output_file
otbdlsal	OTB	N	N/A	4	ordupd	N/A	daily	R	otbdlsal userid/passwd output_file
otbdndld	OTB	N	N/A	4	ordupd	N/A	daily	R	otbdndld userid/passwd output_file
otbprg	OTB	N	N/A	ad hoc	N/A	N/A	monthly	N	otbprg userid/passwc
otbupld	OTB	Y	File-based	ad hoc	N/A	N/A	daily	R	otbupld userid/passwd input_file reject_file
otbupld	OTB	Y	File-based	ad hoc	N/A	N/A	daily	R	otbupld userid/passwd input_file reject_file
poscndld	Point of Sale Interface	N	N/A	4	posndld	prepost poscndld pos	daily	R	poscndld userid/passwd outputfil
posndld	Point of Sale Interface	Y	Store	ad hoc	N/A	prepost posndld pos	daily	R	posndld userid/passwd output_filename
posgpdld	Point of Sale Interface	N	N/A	4	reclsdly	N/A	daily	R	posgpdld userid/passwd output_file
posrefresh	Point of Sale Interface	N	N/A	ad hoc	N/A	N/A	daily	N	posrefresh userid/passwd output_file store_n
posupld	Sales	Y	File-based	2	saexprms(ReSA)	prepost posupld post	daily	R	posupld userid/passwd infle rejfile vatfile itemfile lockfile
precostcalc	Deals	Y	Supplier	2	disinst	prepost precostcalc pre	daily	R	precostcalc userid/passwd supplier (May use the batch_precostcalc.ksh for launching this program as it is created base
reclsdly	Pre/post functionality	N	N/A	all phases	N/A	costcalc	daily	N	on performance considerations)
reclsdly	Item Maintenance	Y	Reclass no	4	cremhierdy	prepost reclsdly pos	daily	R	prepost userid/passwd program pre_or_pos
repladj	Replenishment	Y	Dept	3	rplupld	reqlst	daily	R	reclsdly userid/passwd process_mode
replsizeprofile	Replenishment	N	N/A	ad hoc	prepost replsizeprofile pre	N/A	ad hoc	N	repladj userid/passwd
reqlst	Replenishment	Y	Partition (Item)	3	posupld	replsizeprofile (if size profiles are used to setup	daily	R	replsizeprofile userid/passwd Y/N (Y/N indicator indicates if allocations is installed or not, if installed pre job for thi
rlmaint	Replenishment	Y	Location	3	rplupld	replenishment)	daily	R	program has to be run prepost replsizeprofile pre)
rplapprv	Replenishment	N	N/A	3	reqlst	prepost rplapprv pre	daily	R	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based
rplathistprg	Replenishment	N	N/A	ad hoc	reqlst	prepost rplathistprg	daily	R	on performance considerations)
rplupld	Replenishment	Y	Location	3	reqlst	prepost rplupld	daily	R	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based
rlbld	Replenishment	Y	Supplier	3	reqlst	prepost rlbld	daily	R	on performance considerations)
rplext	Replenishment	Y	Dept	3	reqlst	prepost rplext	daily	R	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based
rplprg	Replenishment	N	N/A	ad hoc	reqlst	prepost rplprg	daily	N	on performance considerations)
rplsplit	Replenishment	Y	Supplier	3	reqlst	prepost rplsplit	daily	R	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based
rplmmovavg	Pricing	Y	Store	3	reqlst	prepost rplmmovavg	daily	R	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based
rtvprg	RTV	N	N/A	ad hoc	reqlst	prepost rtvprg	daily	N	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based
sacrypt	Sales Audit	Y	Store/Day	SA	reqlst	prepost sacrypt	daily	N	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based
saescheat	Sales Audit	N	N/A	SA	reqlst	prepost saescheat	monthly	R	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based
saexpach	Sales Audit	N	N/A	SA	reqlst	prepost saexpach	daily	R	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based
saexpgl	Sales Audit	N	N/A	SA	reqlst	prepost saexpgl	daily	R	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based
saexpim	Sales Audit	N	N/A	SA	reqlst	prepost saexpim	daily	R	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based
saexpdwd	Sales Audit	Y	Store	SA	reqlst	prepost saexpdwd	daily	R	reqlst userid/passwd partition_position (May use the batch_reqlst.ksh for launching this program as it is created based

saexprms	Sales Audit	Y	Store	SA	satotals sarules sapreexp satotals sarules sapreexp	saprepost saexprms post	daily	R	saexprms user/passwd
saexpuar	Sales Audit	N	N/A	SA	satotals sapreexp	N/A	daily	R	saexpuar user/passwd sagetref user/passwd itemfile wastefile ref_itemfile prim_variantfile varupcfile storedayfile codesfile errorfile ccvalf storeposfile tendertypefile merchcodefile partnerfile supplierfile employeefile bannerfile. (To prevent a file from being written, place a '-' in its place. Note: Item files must all be written together).
sagetref saimpadj	Sales Audit Sales Audit	N N	N/A N/A	SA SA	sastdycr saimptlogfin	saimptlog satotals saprepost saimptlog pos (Use sql Loader to load data into ReSA tables)	daily daily	R R	saimpadj user/passwd input_file rej_fil
saimptlog	Sales Audit	Y	Store/Day	SA	sagetref saprepost saimptlog pre saimptlog savouch salstage filgldn1	daily	N	saimptlog user/pw infile badfile itemfile wastefile refitemfile primvariantfile varupcfile storedayfile promfile codesfile errorfile ccvalfile storeposfile tendertypefile merchcodefile partnerfile supplierfile employeefile bannerfile	
saimptlogfin	Sales Audit	N	N/A	SA	saimptlog savouch salstage filgldn1	satotals	daily	R	saimptlogfin user/passwd store_day_file
salapnd saldly saleoh salns salmaint	Stock Ledger Stock Ledger Stock Ledger Sales Stock Ledger	N Y Y N N	N/A Store/Wh Y Dept N/A N/A	3 3 3 0 ad hoc	filgldn2 salstage salmth N/A N/A N/A salweek pre_dw_extract.kah(RMS to RDW RETL Extract)	N/A salweek N/A N/A N/A	daily daily half yearly daily half yearly	R R N R N	salapnd user/passwd saldly user/passwc saleoh user/passwc salns user/passwc salmaint user/passwd pre_or_post
salmth salprg	Stock Ledger Stock Ledger	Y N	Dept N/A	3 ad hoc	N/A	prepost salmth post N/A saldly salapnd salweek dealfct rpmmovavg filgldn1 filgldn2	monthly daily	R N	salmth user/passwd salprg user/passwc
salstage	Stock Ledger	N	N/A	3	posupld saldly stkdy salapnd prepost salweek pre dealfct dealinc vendinv vendinvf	daily	N	salstage user/passwd	
salweek sapreexp saprepost	Stock Ledger Sales Audit Sales Audit	Y N N	Dept N/A N/A	3 SA SA	vendinv SA audit process N/A saprepost sapurge pre (This program should be run as the last program in the ReSA batch schedule)	salmth prepost salweek post (Before any SA export process N/A	weekly daily daily	R R N	salweek user/passwd sapreexp user/passwc saprepost user/passwd program pre_or_pos
sapurge	Sales Audit	Y	Store	SA		saprepost sapurge post sapreexp saescheat	daily	R	sapurge user/passwd deleted_items_file [optional list of store days to be deleted]
sarules	Sales Audit	N	N/A	SA	satotals (It should run before the DTESYS batch program and before the next store/day's transactions are received)	sapreexp saescheat	daily	R	sarules user/passwd store_no
sastdycr satotals savouch sccext schedprg slmain soutdrld stkdy stkprg stkschedxpld	Sales Audit Sales Audit Sales Audit Costing Organizational Hierarchy Item Maintenance Forecasting Stock Ledger Stock Ledger Stock Ledger	N N N Y N N Y N N Y	N/A N/A N/A Cost change N/A N/A Domain Id Dept N/A Location	date_set SA SA 3 ad hoc ad hoc 4 3 ad hoc 0	date_set saimptlogfin saimptlog (and its SQL Load process cstisldex.kah (RMS to RDW RETL extract) N/A lcrbid N/A stkvar N/A prepost stkupld pre	dtseys sarules saimptlogfin prepost sccext post N/A N/A N/A salweek prepost stkprg post stkupld	daily daily daily daily monthly ad hoc daily monthly daily	R R R R R R N R R	sastdycr user/passwd [YYYYMMDD] satotals user/passwd store_no savouch user/passwd infile rejfile tendertype_fil sccext user/passwc schedprg user/passwc slmain user/passwd soutdrld user/passwd stkdy user/passwd stkprg user/passwc stkschedxpld user/passwc
stkupld stkupld stkvar	Stock Ledger Stock Ledger Stock Ledger	Y Y Y	Location Dept Dept	3 1 1	stkupld lftskup N/A	prepost stkupld post N/A N/A	daily daily daily	R R R	stkupld user/passwd stkupld user/passwd input_file reject_fil stkvar user/passwd [report_file_name
stkupld stgldnld storeadd supcnstr supmth tamperctn tkctdrld tftposdn tranupld tftprg tftposdn txrtupld vatdxpl	Stock Ledger Stock Ledger Maintenance - Location Replenishment Stock Ledger Receiving Maintenance Sales Tax Trade Management Transfers Point of Sale Interface Sales Tax Maintenance - VAT	Y Y N N Y N N N N N N N Y	Dept Dept N/A N/A Dept N/A N/A N/A File-based N/A N/A N/A Vat Region	3 4 ad hoc 3 3 ad hoc ad hoc 4 ad hoc ad hoc 4 0	stkschedxpld wasteadj N/A N/A rplbld N/A N/A txrposdn N/A N/A N/A N/A N/A	stkupld wasteadj N/A N/A rplbld N/A N/A prepost tftposdn pos N/A N/A tftposdn N/A N/A prepost vatdxpl pos	daily weekly daily daily monthly ad hoc daily daily daily daily ad hoc daily	R R R R R R R R R R R R	stkupld user/passwd stgldnld user/passwd input_file storeadd user/passwc supcnstr user/passwd supmth user/passwd tamperctn user/passwd tkctdrld user/passwd filename print_online_ind days_in_advance [locator tftposdn user/passwd output_fil tranupld user/passwd infile tftprg user/passwd txrposdn user/passwc txrtupld username/passwd input_file reject_fil vatdxpl user/passwc
vendinv	Deals	Y	Deal Id	3	dealfact salstage(if daily) prepost vendinv pre	stkupld wasteadj salweek(if weekly) salmth (if monthly)	daily	R	vendinv user/passwd
vendinvf vrpbld	Deals Replenishment	Y Y	Deal Id Supplier	3 2	salstage(if daily) prepost vendinvf pre ediupack	prepost vendinvf pos salmth (if monthly) prepost vrpbld pos	daily daily	R R	vendinvf user/passwd vrpbld user/passwd
wasteadj whadd	Stock Ledger Maintenance - Location	Y N	Store N/A	3 ad hoc	N/A (Must be run after all replenishment batch programs).	stkupld stkupld N/A	daily daily	R R	wasteadj user/passwd whadd user/passwd
whstrasg	Maintenance - Location	N	N/A	3		prepost whstrasg post	daily	R	whstrasg user/passwd

RPM Dependency and Scheduling Details									
Program Name	Functional Area	Threaded	Driver	Phase	Pre-dependency	Post-dependency	Timing	Uses Restart/Recovery	Run Parameters for Programs
ItemReclassBatch	Future Retail	N	N/A	N/A	recdsdy(RMS)	NewItemLocBatch	daily/ad hoc	N	itemReclassBatch.sh rpm-app-userid password
NewItemLocBatch	Future Retail	N	N/A	N/A	storeadd(RMS), ItemReclassBatch	LocationMoveBatch	daily/ad hoc	N	NewItemLocBatch.sh rpm-app-userid password [status [error-commit-count]
LocationMoveBatch	Zone Structure/Future Retail	Y	Location move	N/A	NewItemLocBatch LocationMoveBatch	PriceEventExecutionBatch	daily	N	locationMoveBatch.sh rpm-app-userid password
PriceEventExecutionBatch	Price Change/Clearance/Promotion	Y	Pricing event	N/A	salstage (RMS) PriceEventExecutionBatch	PriceEventExecutionRMSBatch	daily	N	priceEventExecutionBatch.sh rpm-app-userid password
PriceEventExecutionRMSBatch	Price Change/Clearance/Promotion	Y	Pricing event	N/A	PriceEventExecutionRMSBatch	PriceEventExecutionDealsBatch	daily	N	priceEventExecutionRMSBatch.sh rpm-app-userid password
PriceEventExecutionDealsBatch	Price Change/Clearance/Promotion	Y	Pricing event	N/A	PriceEventExecutionRMSBatch	MerchExtractKickOffBatch	daily	N	priceEventExecutionDealsBatch.sh rpm-app-userid password
PriceStrategyCalendarBatch	Price Strategy	N	Price strategy	N/A	N/A	MerchExtractKickOffBatch	daily	N	priceStrategyCalendarBatch.sh rpm-app-userid password
WorksheetAutoApproveBatch	Pricing Worksheet	Y	Price strategy	N/A	PriceEventExecutionBatch storeadd (RMS)	MerchExtractKickOffBatch	daily	N	worksheetAutoApproveBatch.sh rpm-app-userid password
MerchExtractKickOffBatch	Pricing Worksheet	Y	Price strategy	N/A	WorksheetAutoApproveBatch PriceStrategyCalendarBatch MerchExtractKickOffBatch	N/A	daily	N	merchExtractKickOffBatch.sh rpm-app-userid password
RPMtoORPOSPublishBatch.sh	Price Change/Clearance/Promotion	N	N/A	N/A	WorksheetAutoApproveBatch	N/A	daily	N	ksh RPMtoORPOSPublishBatch.sh <userid/passwd@sid> <log-path> <error-path>
RPMtoORPOSPublishExport.sh	Price Change/Clearance/Promotion	Y	Location	N/A	RPMtoORPOSPublishBatch.sh	N/A	daily	N	ksh RPMtoORPOSPublishExport.sh <userid/passwd@sid> <Numberof slots> <logpath> <error path> <Export path>
RegularPriceChangePublishBatch	Regular Price Changes	Y	Price event (item/loc)	N/A	WorksheetAutoApproveBatch	RegularPriceChangePublishExport	daily/ad hoc	N	regularPriceChangePublishBatch.sh rpm-app-userid password
RegularPriceChangePublishExport	Regular Price Changes	N	Price event (item/loc)	N/A	RegularPriceChangePublishBatch	RegularPriceChangePublishExport	daily/ad hoc	N	regularPriceChangePublishExport.sh rpm-db-userid/pwd@database [export-path]
ClearancePriceChangePublishBatch	Clearances	Y	Price event (item/loc)	N/A	WorksheetAutoApproveBatch	ClearancePriceChangePublishExport	daily/ad hoc	N	clearancePriceChangePublishBatch.sh rpm-app-userid password
ClearancePriceChangePublishExport	Clearances	Y	Price event (item/loc)	N/A	ClearancePriceChangePublishBatch	ClearancePriceChangePublishExport	daily/ad hoc	N	clearancePriceChangePublishExport.sh rpm-db-userid/pwd@database [export-path]
PromotionPriceChangePublishBatch	Promotions	Y	Price event (item/loc)	N/A	WorksheetAutoApproveBatch	PromotionPriceChangePublishExport	daily/ad hoc	N	promotionPriceChangePublishBatch.sh rpm-app-userid password
PromotionPriceChangePublishExport	Promotions	N	Price event (item/loc)	N/A	PromotionPriceChangePublishBatch	N/A	daily/ad hoc	N	promotionPriceChangePublishExport.sh rpm-db-userid/pwd@database [export-path]
PriceChangeAutoApproveResultsPurgeBatch	Purge	N	N/A	N/A	N/A	N/A	daily/ad hoc	N	priceChangeAutoApproveResultsPurgeBatch.sh rpm-app-userid password
PriceChangePurgeBatch	Purge	N	N/A	N/A	N/A	N/A	daily/ad hoc	N	priceChangePurgeBatch.sh rpm-app-userid password
PriceChangePurgeWorkspaceBatch	Purge	N	N/A	N/A	N/A	N/A	daily/ad hoc	N	priceChangePurgeWorkspaceBatch.sh rpm-app-userid password
PromotionPurgeBatch	Purge	N	N/A	N/A	N/A	N/A	daily/ad hoc	N	promotionPurgeBatch.sh rpm-app-userid password
PurgeExpiredExecutedOrApprovedClearancesBatch	Purge	N	N/A	N/A	N/A	N/A	daily/ad hoc	N	purgeExpiredExecutedOrApprovedClearancesBatch.sh rpm-app-userid password
PurgeUnusedAndAbandonedClearancesBatch	Purge	N	N/A	N/A	N/A	N/A	daily/ad hoc	N	purgeUnusedAndAbandonedClearancesBatch.sh rpm-app-userid password
PurgeLocationMovesBatch	Purge	N	N/A	N/A	N/A	N/A	daily/ad hoc	N	purgeLocationMovesBatch.sh rpm-app-userid password
ZoneFutureRetailPurgeBatch	Purge	N	N/A	N/A	N/A	N/A	ad hoc	N	zoneFutureRetailPurgeBatch.sh rpm-app-userid password
ItemLocDeleteBatch	Purge	N	N/A	N/A	N/A	N/A	ad hoc	N	itemLocDeleteBatch.sh rpm-app-userid password
priceChangeAreaDifferentialBatch	Price Change	Y	N/A	N/A	N/A	N/A	ad hoc	N	priceChangeAreaDifferentialBatch rpm-app-userid password
refreshPosDataBatch	Point of Sale Interface	Y	N/A	N/A	N/A	N/A	ad hoc	N	RefreshPosDataBatch <username> <password> <location>[date (YYYYMMdd]
PurgePayloadsBatch	Purge	N	N/A	N/A	RPMtoORPOSPublishExport RegularPriceChangePublishExport ClearancePriceChangePublishExport PromotionPriceChangePublishExport	N/A	daily/ad hoc	N	purgePayloadsBatch.sh<rpm-db-userid/password@database>-publish-status-<log-path>

ReIM Dependency and Scheduling Details									
Program Name	Functional Area	Threaded	Driver	Phase	Pre-dependency	Post-dependency	Timing	Uses Restart/Recovery	Run Parameters for Programs
AutoMatch	Invoice Matching (ReIM)	Y	N/A	6	N/A	ReasonCodeActionRollup	daily	R	AutoMatch userid/password
BatchPurge	Invoice Matching (ReIM)	N	N/A	0	N/A	ResolutionPosting	daily	R	BatchPurge userid/password
ComplexDealUpload	Invoice Matching (ReIM)	Y	N/A	5	vendinvct(RMS), vendinv(RMS)	AutoMatch	daily	R	ComplexDealUpload userid/password BlockSize PartitionNo
DiscrepancyPurge	Invoice Matching (ReIM)	N	N/A	1	N/A	N/A	daily	R	DiscrepancyPurge userid/password
DisputedCreditMemoRollup	Invoice Matching (ReIM)	N	N/A	6	ReasonCodeActionRollup	ResolutionPosting	daily	R	DisputedCreditMemoRollup userid/password
EdlinvoiceUpload	Invoice Matching (ReIM)	Y	N/A	5	eddlinv(RMS)	AutoMatch	daily	R	EdlinvoiceUpload userid/password "EDI input file with path" "EDI reject file with path"
EdlinvoiceDownload	Invoice Matching (ReIM)	N	N/A	7	ResolutionPosting	N/A	daily	R	EdlinvoiceDownload userid/password
FixedDealUpload	Invoice Matching (ReIM)	Y	N/A	5	vendinvct(RMS), vendinv(RMS)	AutoMatch	daily	R	FixedDealUpload userid/password BlockSize PartitionNo
ReasonCodeActionRollup	Invoice Matching (ReIM)	N	N/A	6	AutoMatch	DisputedCreditMemoRollup	daily	R	ReasonCodeActionRollup userid/password
ReceiptWriteoff	Invoice Matching (ReIM)	N	N/A	6	AutoMatch	N/A	daily	R	ReceiptWriteoff userid/password
ResolutionPosting	Invoice Matching (ReIM)	N	N/A	6	ReasonCodeActionRollup, DisputedCreditMemoRollup	N/A	daily	R	ResolutionPosting userid/password

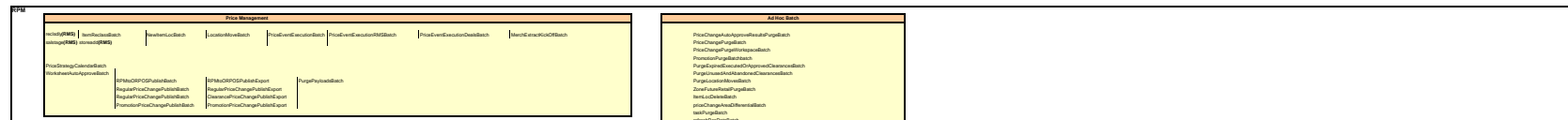
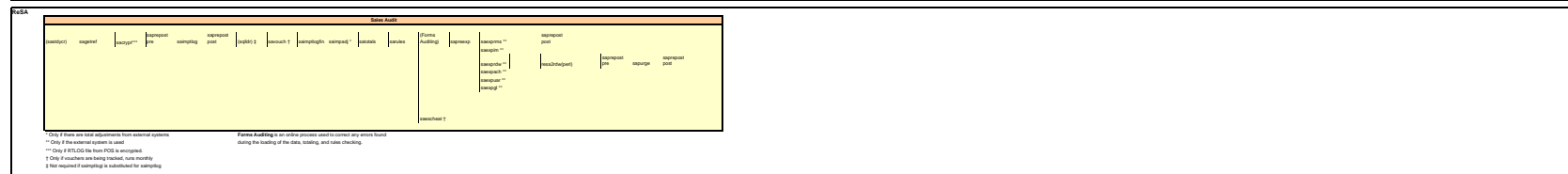
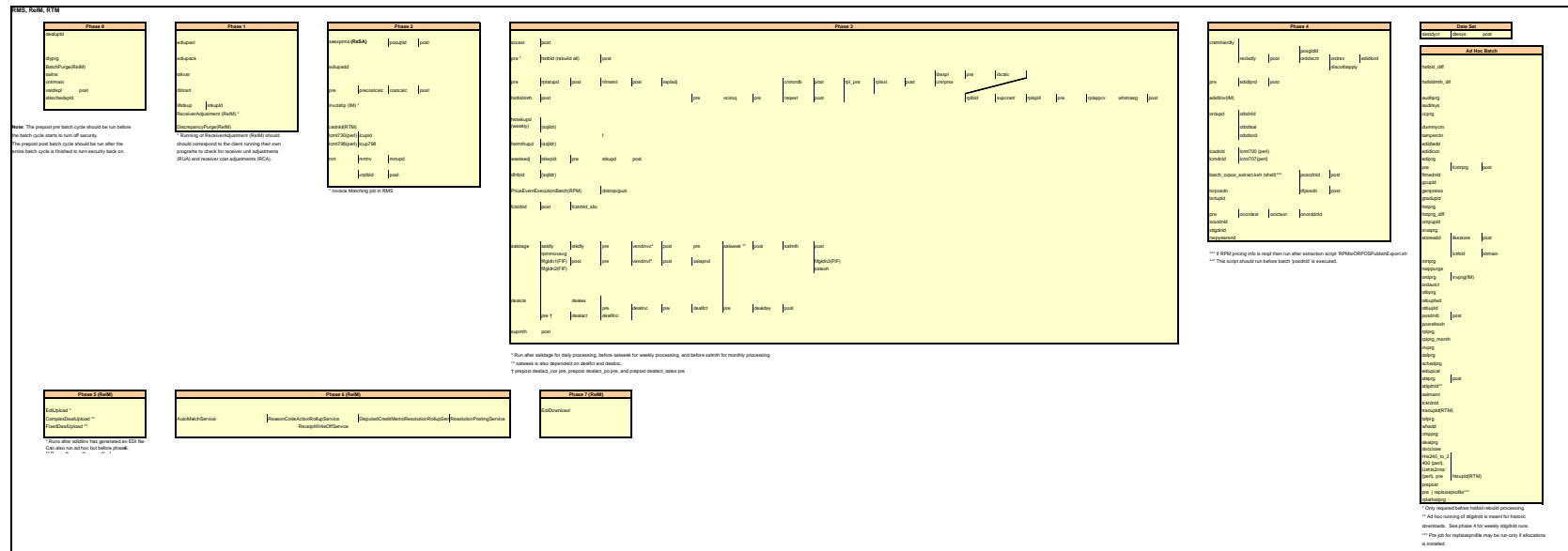
RMS to RPAS RETL Extracts Dependency and Scheduling Details (EXTRACTS FOR RPAS)									
Program Name	Functional Area	Threaded	Driver	Phase	Pre-dependency	Post-dependency	Timing	Uses Restart/Recovery	Run Parameters for Programs
pre_rmse_rpas.ksh	Planning/Forecast System Interface	N	N/A	N/A	N/A. This is a pre setup script	N/A	daily	N	N/A
rmse_rpas.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh. (This is the launch script to run the extracts)	Refer to RPAS Operations guide	daily	N	N/A
rmse_rpas_attributes.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh	Refer to RPAS Operations guide	daily	N	N/A
rmse_rpas_daily_sales.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh	Refer to RPAS Operations guide	daily	N	N/A
rmse_rpas_domain.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh stmain recdsdy dlyprg	Refer to RPAS Operations guide	daily	N	N/A
rmse_rpas_item_master.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh recdsdy dlyprg	Refer to RPAS Operations guide	daily	N	N/A
rmse_rpas_merchier.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh dlyprg	Refer to RPAS Operations guide	daily	N	N/A
rmse_rpas_orghier.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh stkdy	Refer to RPAS Operations guide	daily	N	N/A
rmse_rpas_stock_on_hand.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh	Refer to RPAS Operations guide	daily	N	N/A

rmse_rpas_store.ksh	Planning/Forecast System Interface	N	N/A	N/A	storeadd dlyprg	pre_rmse_rpas.ksh	Refer to RPAS Operations guide	daily	N	N/A
rmse_rpas_suppliers.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh hstwkupd salweek	pre_rmse_rpas.ksh	Refer to RPAS Operations guide	daily	N	N/A
rmse_rpas_weekly_sales.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh whadd dlyprg	pre_rmse_rpas.ksh	Refer to RPAS Operations guide	daily	N	N/A
rmse_rpas_wh.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh	pre_rmse_rpas.ksh	Refer to RPAS Operations guide	daily	N	N/A
rmsl_rpas_forecast.ksh	Planning/Forecast System Interface	N	N/A	N/A	pre_rmse_rpas.ksh	pre_rmse_rpas.ksh	Refer to RPAS Operations guide	daily	N	rmsl_rpas_forecast.ksh daily or weekly
rmsl_rpas_update_retl_date.ksh	Planning/Forecast System Interface	N	N/A	N/A	After all RMS/Planning System Integration RETL scripts are run		Refer to RPAS Operations guide	daily	N	rmsl_rpas_update_retl_date.ksh CLOSED_ORDER or RECEIVED_QTY

**RMS to RDW RETL Extracts Dependency and Scheduling
Details (EXTRACTS FOR RDW)**

Program Name	Functional Area	Threaded	Driver	Phase	Pre-dependency	Post-dependency	Timing	Uses Restart/Recovery	Run Parameters for Programs
cdedtlx.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
cmptrex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
cmpttrfmx.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
cmpttrfoex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
cmrcycdex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
cmpliyex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
orgaraex.ksh	RDW interface	N	N/A	N/A	A, B, storeadd (RMS), dlyprg (RMS), lcltbl (RMS)	Refer to RDW operations guide	daily	N	N/A
orgchanex.ksh	RDW interface	N	N/A	N/A	A, B, storeadd (RMS), dlyprg (RMS), lcltbl (RMS)	Refer to RDW operations guide	daily	N	N/A
orgchnex.ksh	RDW interface	N	N/A	N/A	A, B, storeadd (RMS), dlyprg (RMS), lcltbl (RMS)	Refer to RDW operations guide	daily	N	N/A
orgdisex.ksh	RDW interface	N	N/A	N/A	A, B, storeadd (RMS), dlyprg (RMS), lcltbl (RMS)	Refer to RDW operations guide	daily	N	N/A
orglrmex.ksh	RDW interface	N	N/A	N/A	A, B, storeadd (RMS), dlyprg (RMS), lcltbl (RMS)	Refer to RDW operations guide	daily	N	N/A
orgloex.ksh	RDW interface	N	N/A	N/A	A, B, storeadd (RMS), dlyprg (RMS), lcltbl (RMS)	Refer to RDW operations guide	daily	N	N/A
orglolex.ksh	RDW interface	N	N/A	N/A	A, B, storeadd (RMS), dlyprg (RMS), lcltbl (RMS)	Refer to RDW operations guide	daily	N	N/A
orgltmex.ksh	RDW interface	N	N/A	N/A	A, B, storeadd (RMS), dlyprg (RMS), lcltbl (RMS)	Refer to RDW operations guide	daily	N	N/A
orgltrex.ksh	RDW interface	N	N/A	N/A	A, B, storeadd (RMS), dlyprg (RMS), lcltbl (RMS)	Refer to RDW operations guide	daily	N	N/A
orggrnex.ksh	RDW interface	N	N/A	N/A	A, B, storeadd (RMS), dlyprg (RMS), lcltbl (RMS)	Refer to RDW operations guide	daily	N	N/A
phasex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
prdclex.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
prdcmpex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
prddepx.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
prddiflex.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
prddivex.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
prddtypex.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
prdgprpx.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
prdislex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
prdlmex.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
prdltmex.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
prdltmlmex.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
prdltmltmex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
prdltmlsmex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
prdpimex.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
prdsbcex.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
prduiaex.ksh	RDW interface	N	N/A	N/A	A, B, cremhierdly (RMS), rectsdly (RMS), dlyprg (RMS)	Refer to RDW operations guide	daily	N	N/A
regnrgpx.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
regrimbex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
rsnex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
seasnex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
subtrantypex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
supctrex.ksh	RDW interface	N	N/A	N/A	A, B, cntnmain (RMS)	Refer to RDW operations guide	daily	N	N/A
supsupex.ksh	RDW interface	N	N/A	N/A	A, B, cntnmain (RMS)	Refer to RDW operations guide	daily	N	N/A
suptrmex.ksh	RDW interface	N	N/A	N/A	A, B, cntnmain (RMS)	Refer to RDW operations guide	daily	N	N/A
suptrpx.ksh	RDW interface	N	N/A	N/A	A, B, cntnmain (RMS)	Refer to RDW operations guide	daily	N	N/A
indrtypex.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A
ltitypx.ksh	RDW interface	N	N/A	N/A	A, B	Refer to RDW operations guide	daily	N	N/A

Integrated Merchandising Batch Schedule



Interface Diagrams for RMS and RPAS

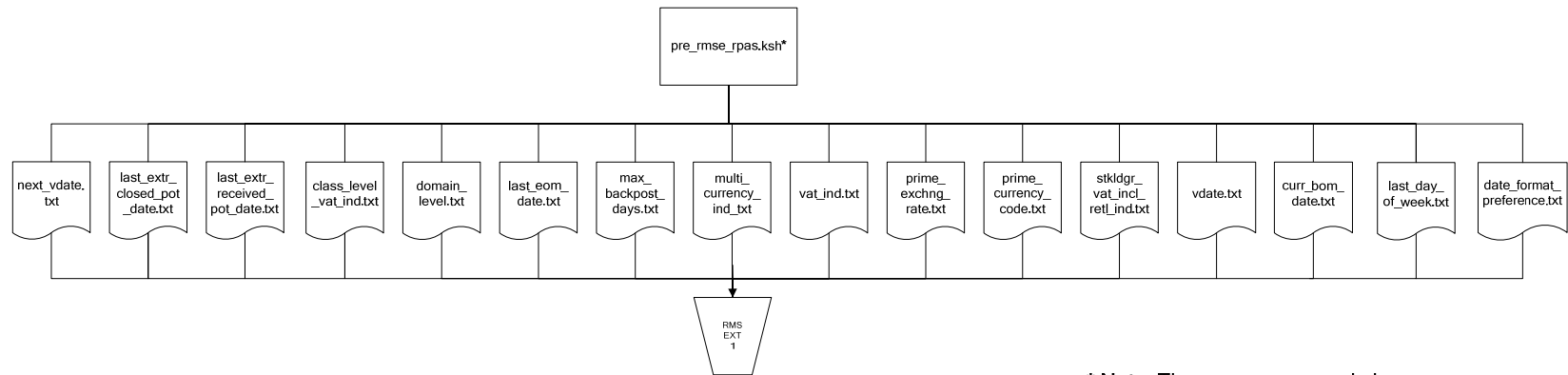
Because RMS is the retailer's central merchandising transactional processing system, it is the principle source of the foundation data needed in some of the Oracle Retail suite of products. RMS provides foundation data to RPAS, and RPAS provides planning data to RMS.

This chapter presents flow diagrams for data processing from sources. The source system's program or output file is illustrated, along with the program or process that interfaces with the source. After initial interface processing of the source, the diagrams illustrate the flow of the data.

Before setting up a program schedule, familiarize yourself with the functional and technical constraints associated with each program. Refer to the Retail Merchandising System Operations Guide for more information about these interface programs.

RMS Pre/Post Extract Diagrams

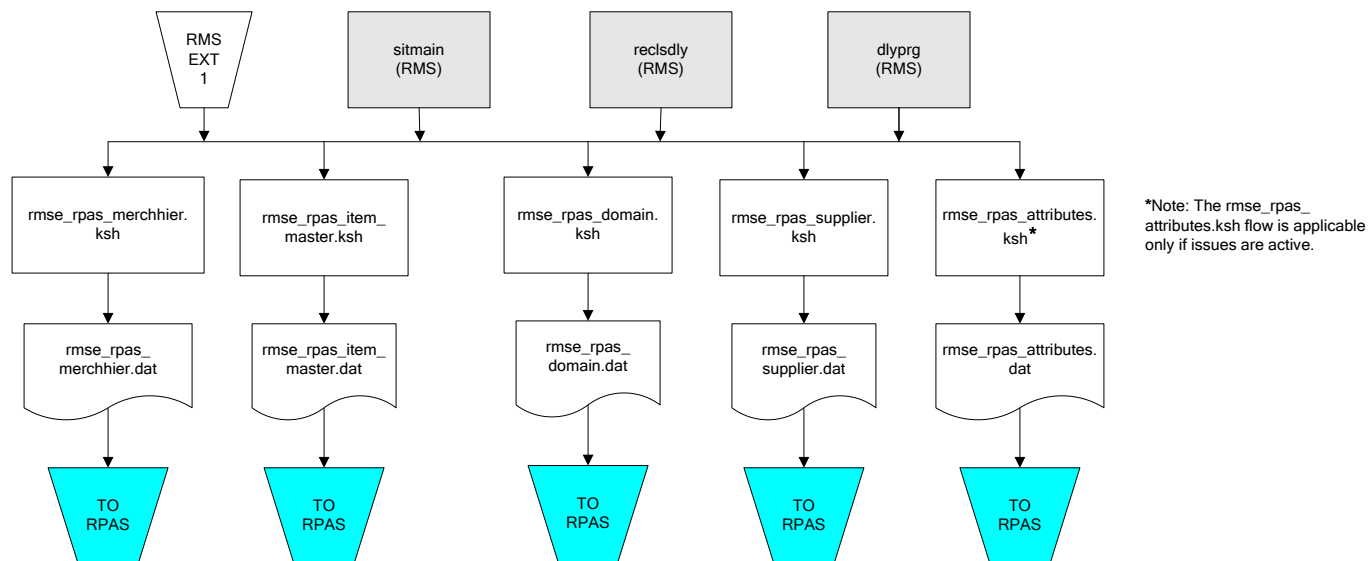
RMS Pre RETL Extract Maintenance



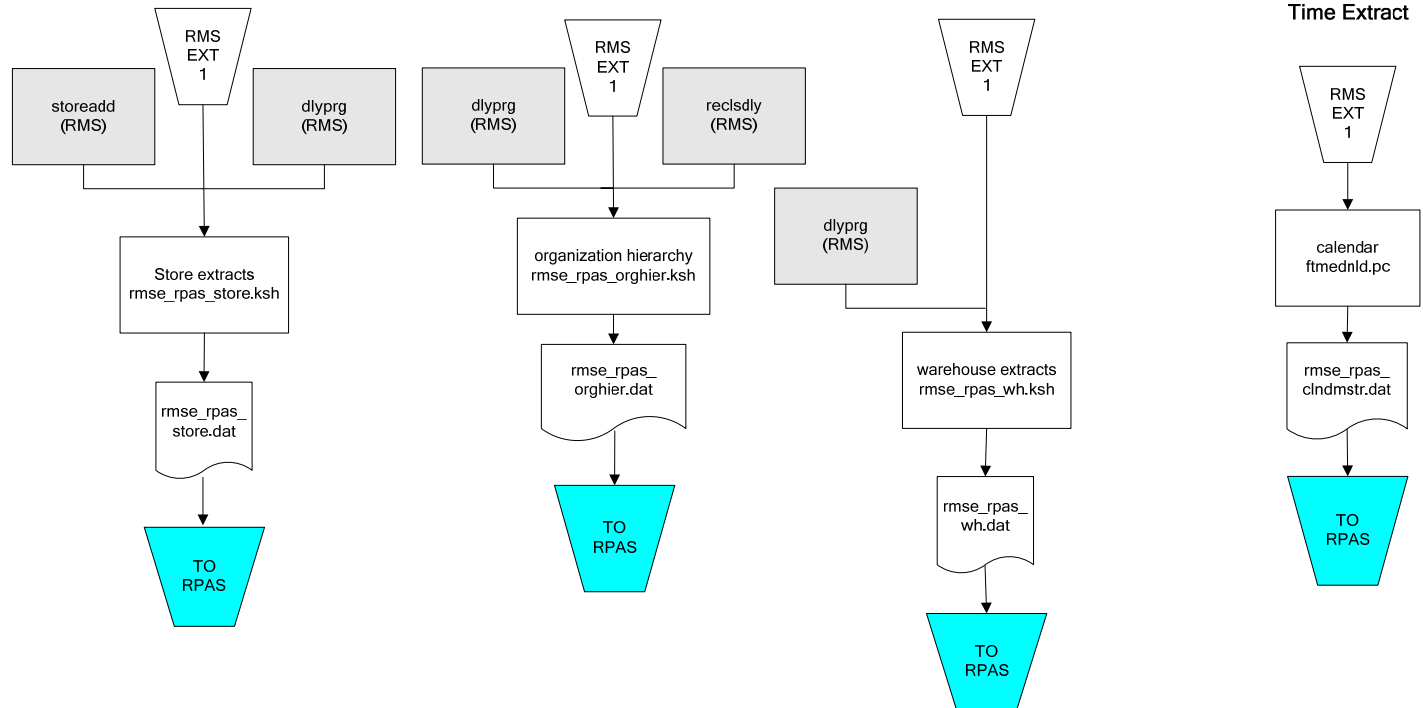
*** Note:** The `pre_rmse_rpas.ksh` program checks for existing .txt output files. Because of this validation, retailers running the program for the first time should include an optional `-c` parameter. This parameter allows the program to run successfully without pre-existing .txt output files.

RMS Foundation Data Extract Diagrams

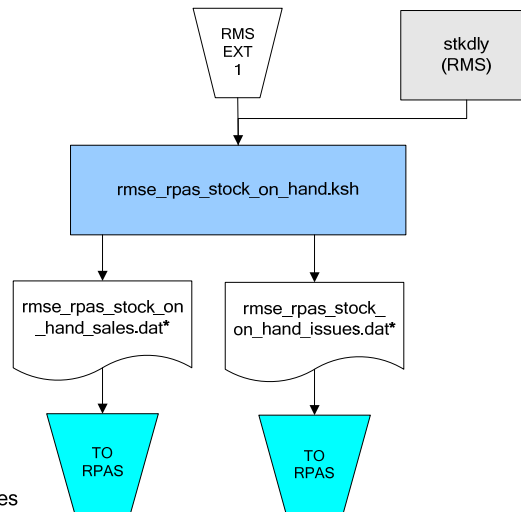
Merchandise Hierarchy for RPAS



Organization Hierarchy for RPAS



RMS Fact Data Extract Diagrams

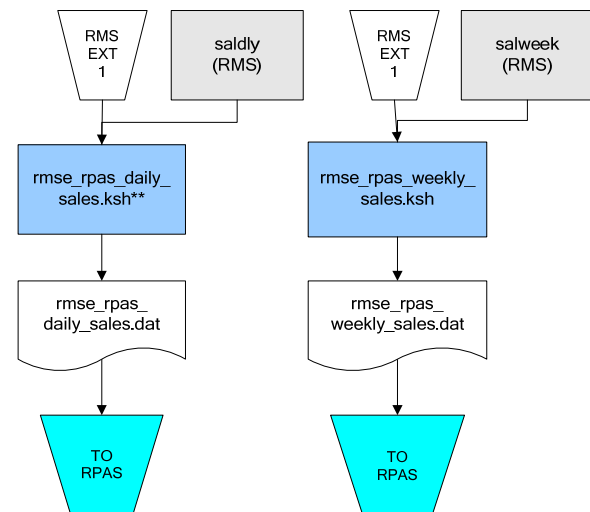


* Note:

If issues are active, the following two files result from the rmse_rpas_stock_on_hand.ksh flow:
 rmse_rpas_stock_on_hand_issues.dat
 rmse_rpas_stock_on_hand_sales.dat

If issues are **not** active, the following file results from the rmse_rpas_stock_on_hand.ksh flow:
 rmse_rpas_stock_on_hand_sales.dat

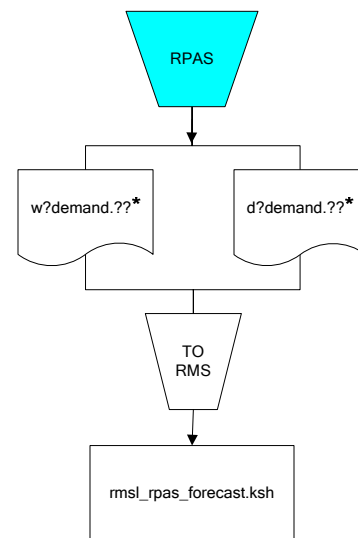
Sales Extracts For RPAS



** Note:

Depending upon the configuration of rmse_rpas_daily_sales.ksh, the data can be pulled from TRAN_DATA_HISTORY or TRAN_DATA.

RPAS-RMS Fact Load Diagram



***Note:**

? can represent the following:

- i (for issues)
- s (for stores)

?? represents domain 01-99.

Interface Diagrams for RMS and RDW

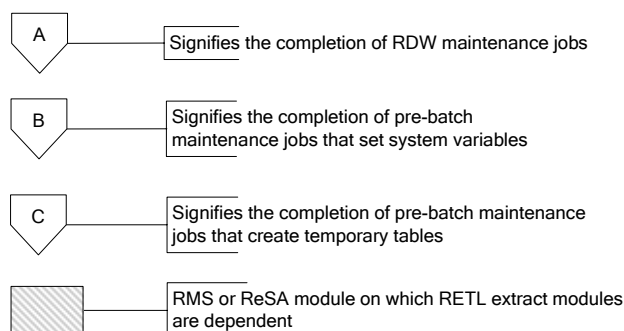
RMS works in conjunction with the Oracle Retail Extract Transform and Load (RETL) framework. RETL provides high-performance processing to extract data from Oracle Retail applications for use in data warehouses. The architecture allows database batch processes to take advantage of parallel processing capabilities.

This chapter presents flow diagrams for the RETL extraction RMS programs. The source system's program or output file is illustrated, along with the program or process that interfaces with the source. Note that the data flows are organized according to the logic (dimension data and table data) of Oracle Retail Data Warehouse (RDW), but you can use the data to suit your business needs.

For detailed information about dimensions and facts, see the Retail Data Warehouse Operations Guide.

For summary information about the configuration, architecture, and features of RETL programs utilized in RMS/ReSA extractions, see the Oracle Retail Management System Operations Guide Volume 3—Backend Configuration and Operations. For more information about the RETL tool, see the current RETL Programmer's Guide.

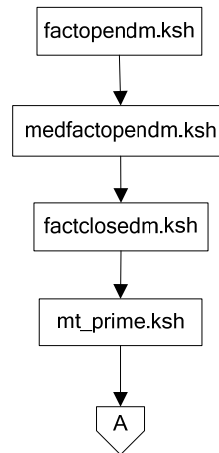
Legend



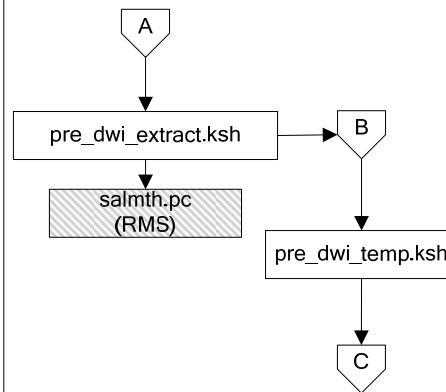
RDW Maintenance

Note:

The modules in this flow are RDW RETL scripts. If the retailer uses RDW, this flow must be completed before starting the pre-batch maintenance flow. If the retailer does not use RDW, these jobs are not required.



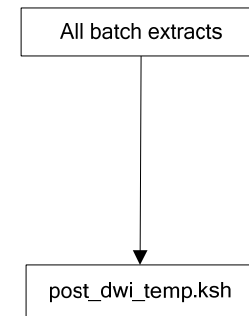
Pre-Batch Maintenance



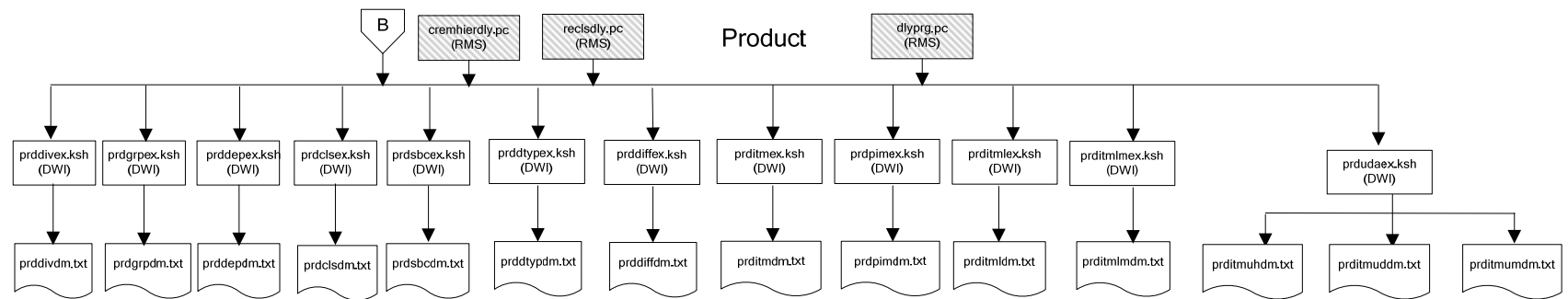
Note:

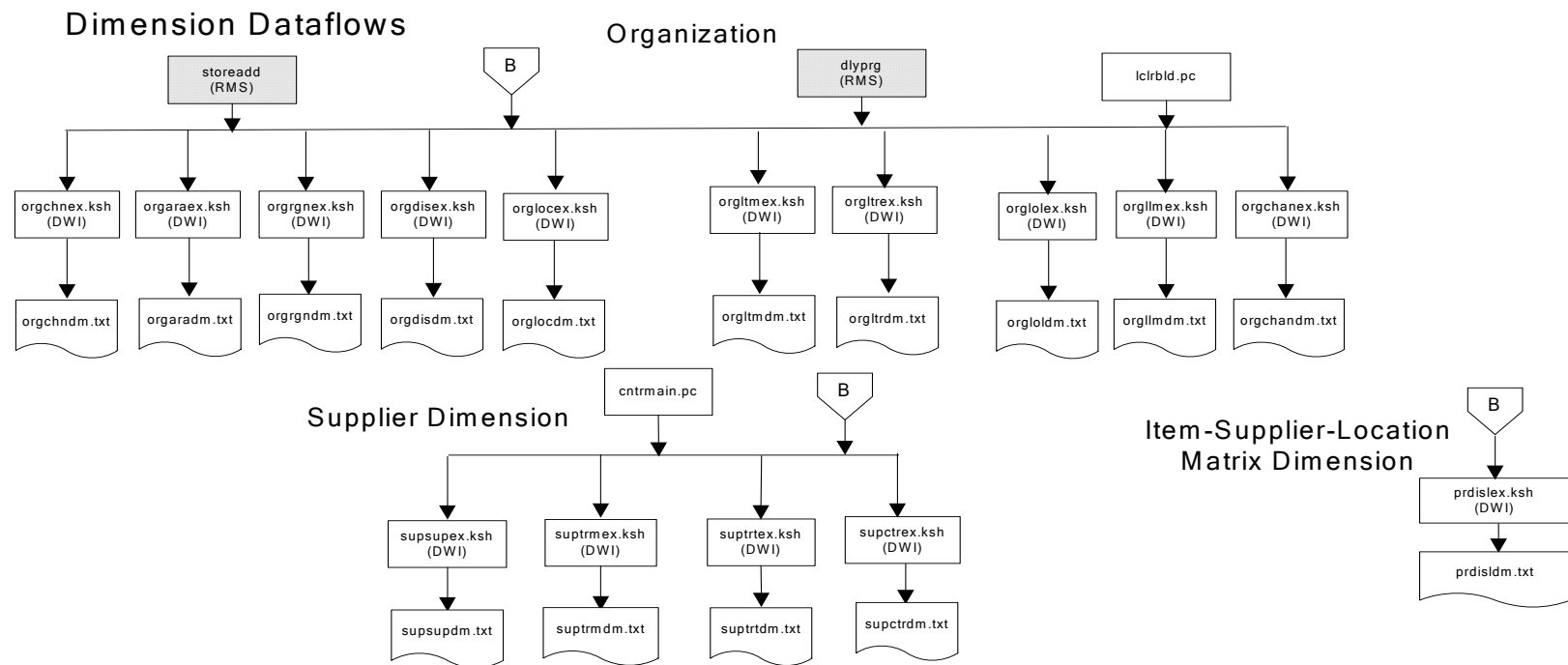
salmth.pc resets the last eom_date. Thus, it must be run after the system indicator is extracted by pre_dwi_extract.ksh.

Post-Batch Maintenance

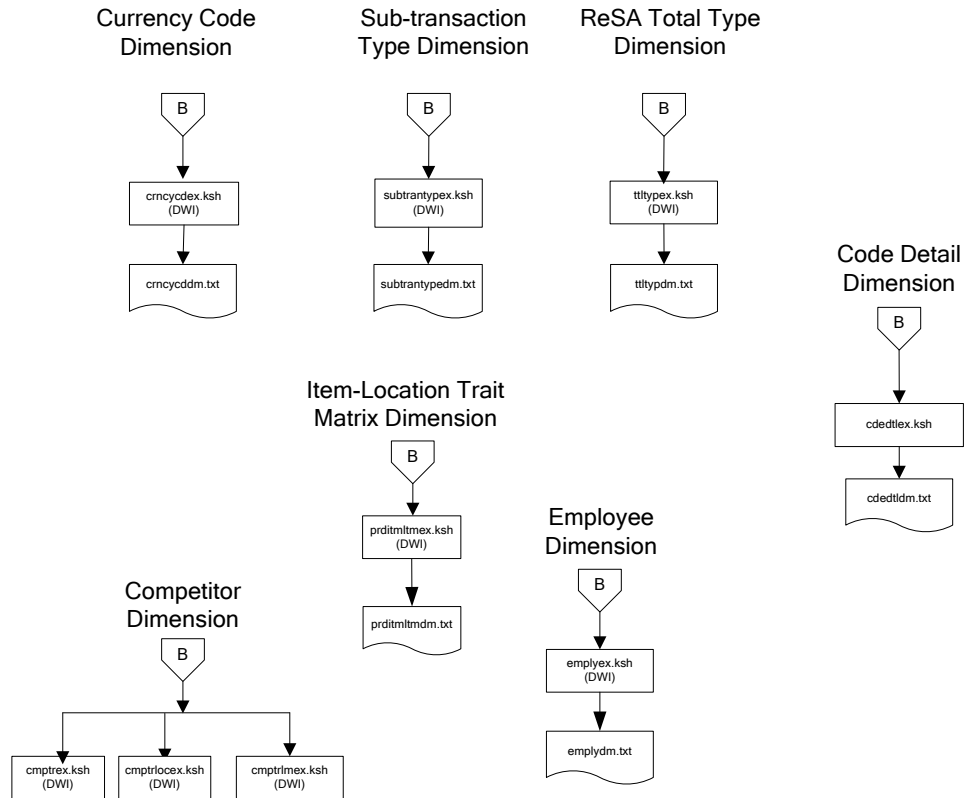


Dimension Dataflows



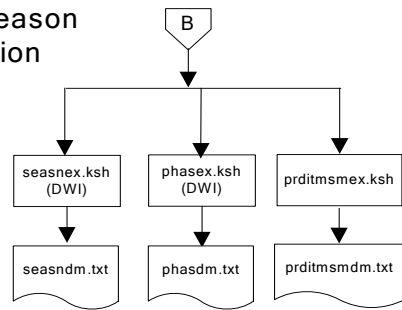


Dimension Dataflows

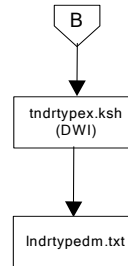


Dimension Dataflows

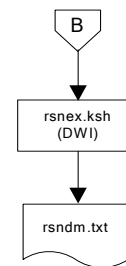
Product Season Dimension



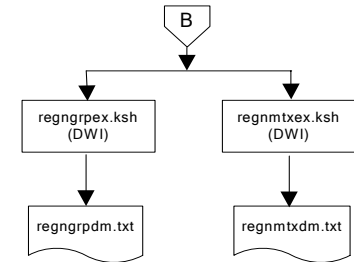
Tender Type Dimension



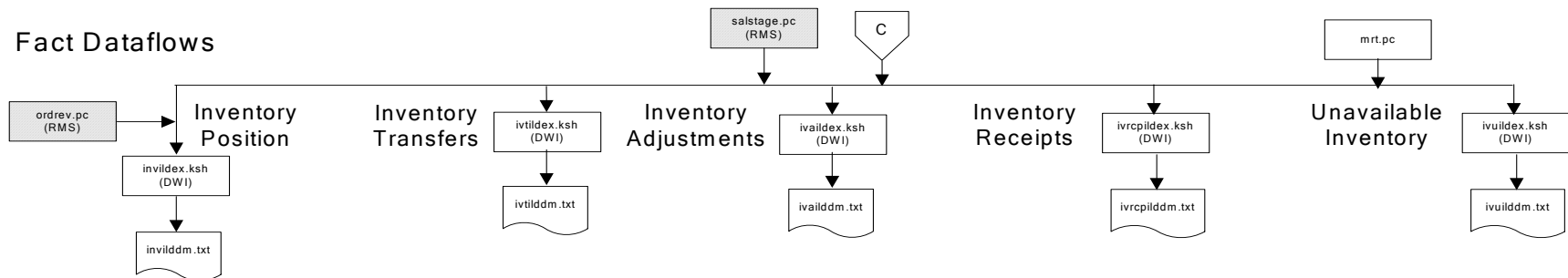
Reason Dimension



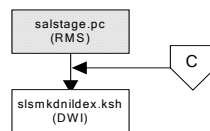
Regionality Dimension



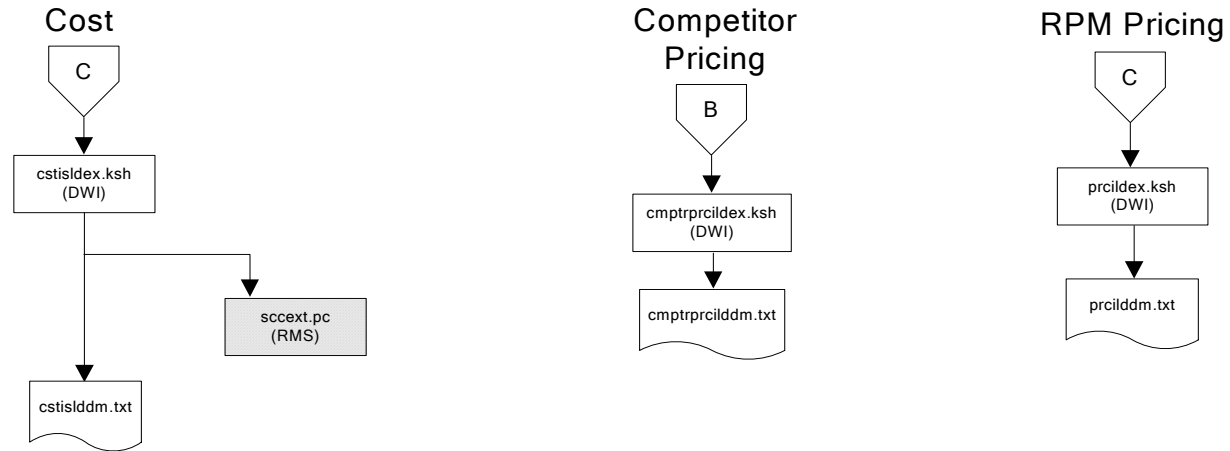
Fact Dataflows



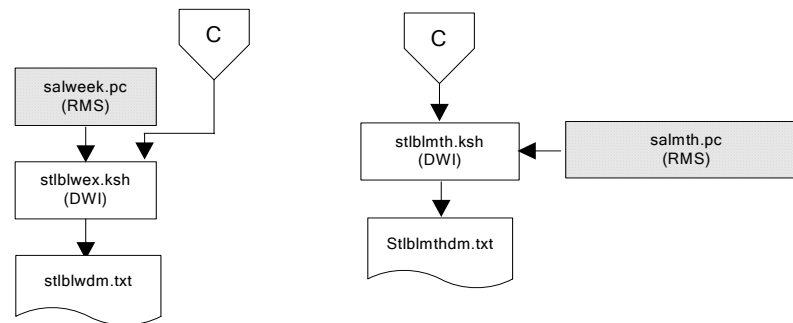
Markdowns



Fact Dataflows



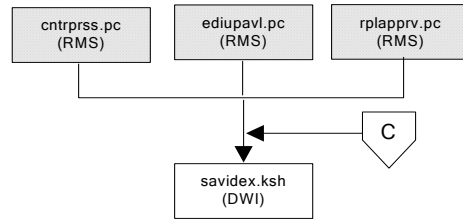
Stock Ledger



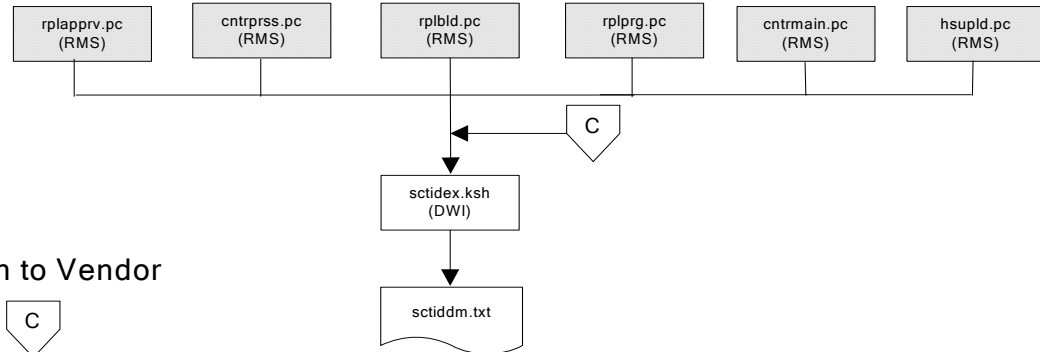
Note:
Run stock ledger fact
loads once weekly.

Fact Dataflows

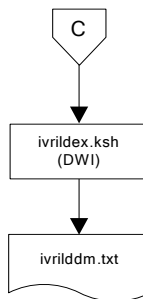
Supplier Availability



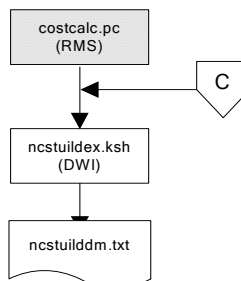
Supplier Contract



Return to Vendor

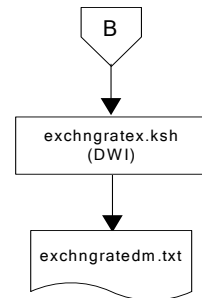


Net Cost

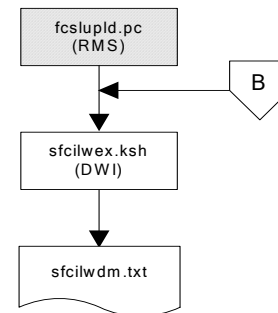


Fact Dataflows

Exchange Rates

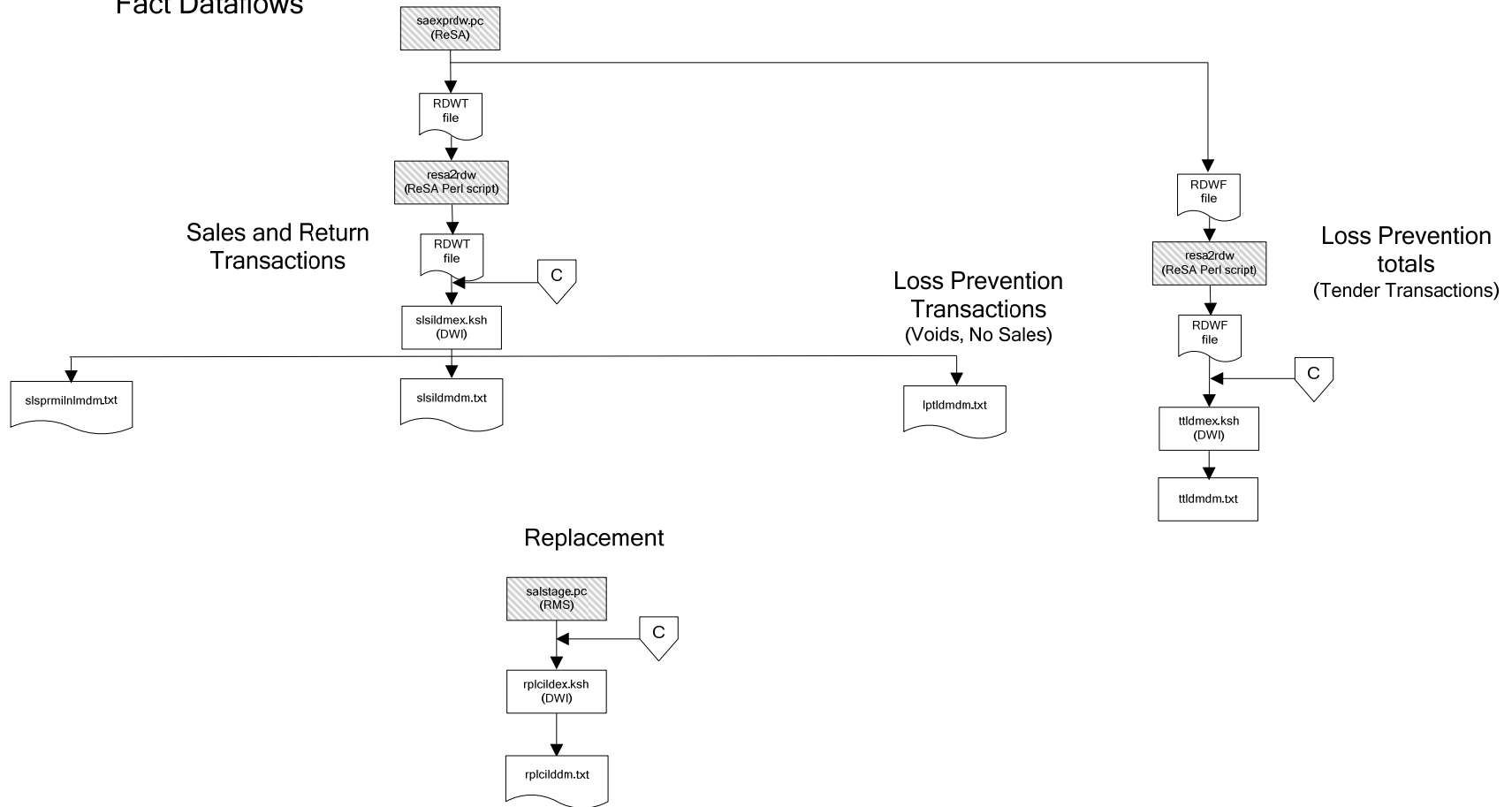


Sales Forecasts

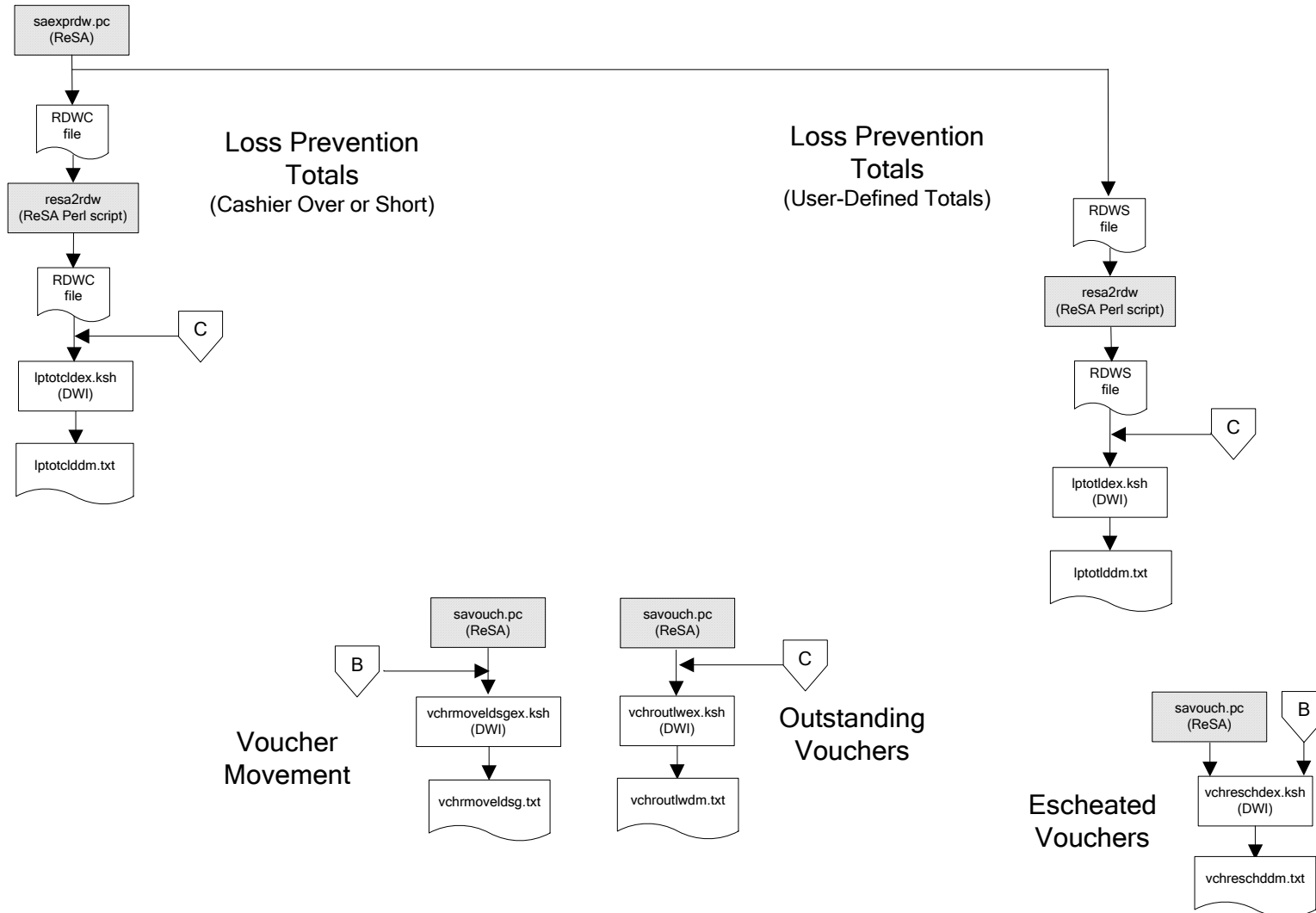


Note:
Run sales forecast fact loads
once weekly.

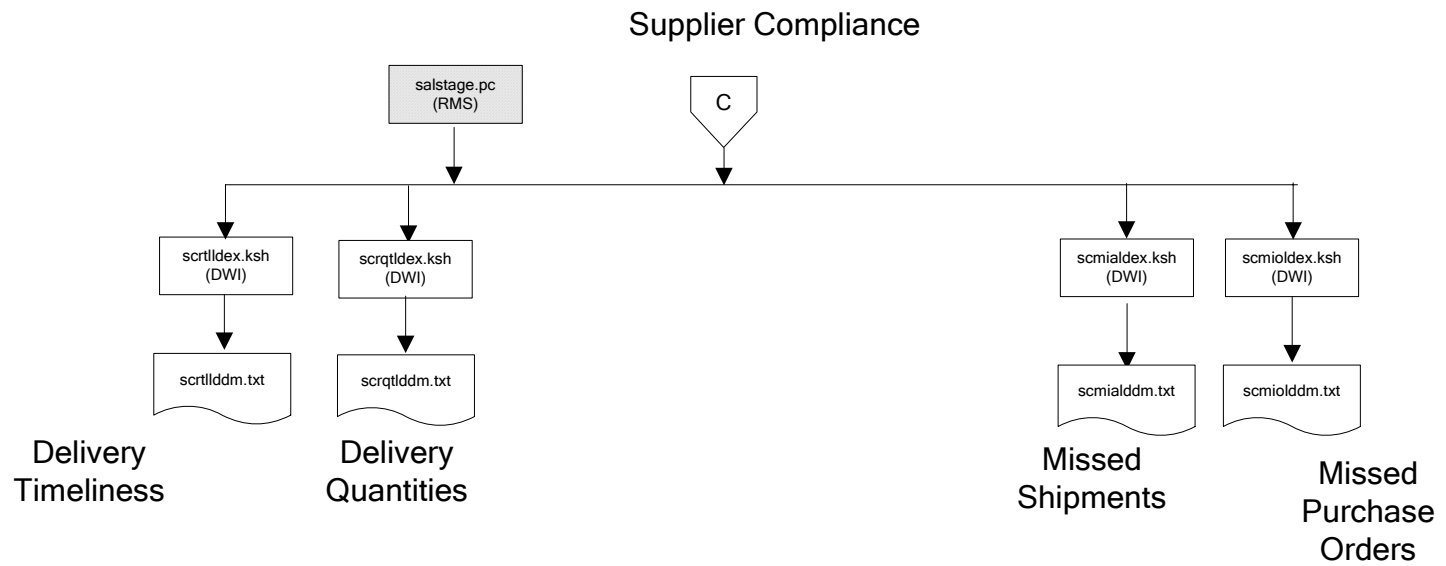
Fact Dataflows



Fact Dataflows



Fact Dataflows



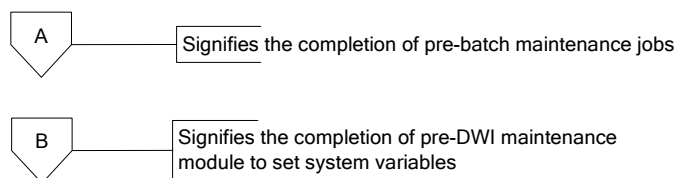
Interface Diagram for RPM and RDW

This following program flow diagram shows the RETL extraction program that extracts the Promotion dimension from RPM through the Data Warehouse Interface (DWI). The diagram shows the output files and the scripts that interface with the source. Note that the outputs are based on the logic (dimension data and table data) of Oracle Retail Data Warehouse (RDW), but you can use the data to suit your business needs.

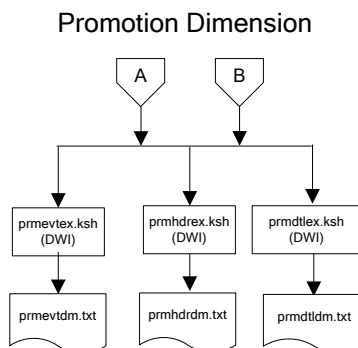
For detailed information about dimensions and facts, see the Retail Data Warehouse Operations Guide.

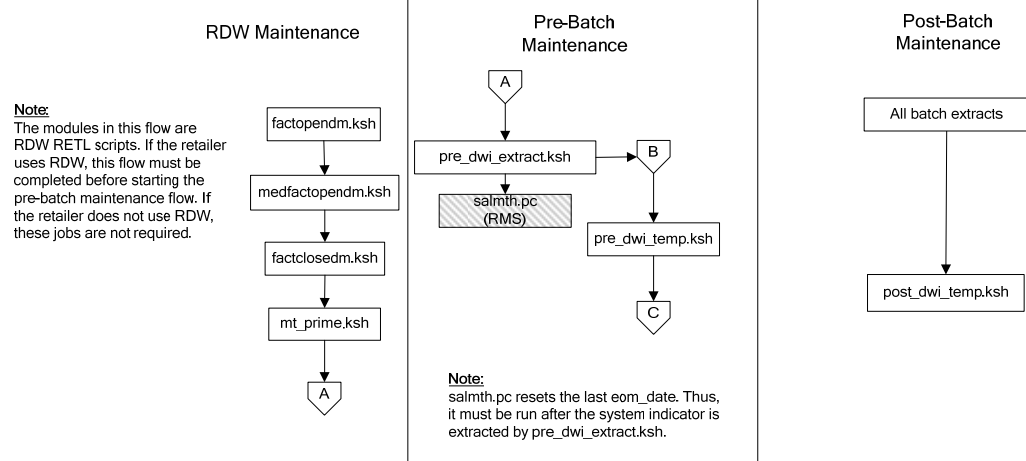
See the Retail Merchandising System Operations Guide Volume 1—Batch Overviews and Designs for more information about the modules shown in the following diagram.

Legend



Program Flow Diagram





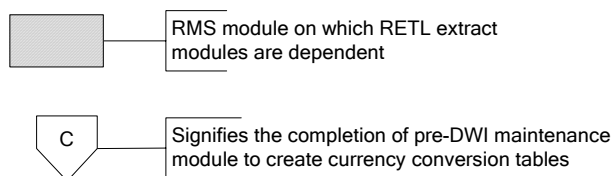
Interface Diagram for ReIM and RDW

This following program flow diagram shows the RETL extraction program that extracts the Promotion dimension from ReIM through the Data Warehouse Interface (DWI). The diagram shows the output files and the scripts that interface with the source. Note that the outputs are based on the logic (dimension data and table data) of Oracle Retail Data Warehouse (RDW), but you can use the data to suit your business needs.

For detailed information about dimensions and facts, see the Retail Data Warehouse Operations Guide.

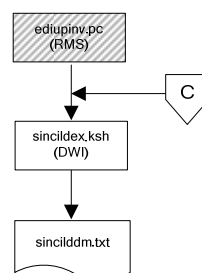
See the Retail Merchandising System Operations Guide Volume 1—Batch Overviews and Designs for more information about the modules shown in the following diagram.

Legend



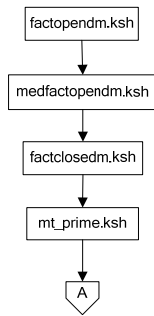
Program Flow Diagram

Supplier Invoice Cost

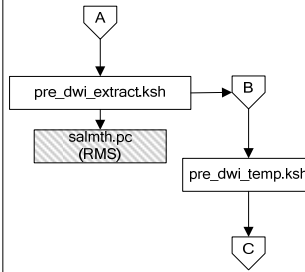


Note:
The modules in this flow are RDW RETL scripts. If the retailer uses RDW, this flow must be completed before starting the pre-batch maintenance flow. If the retailer does not use RDW, these jobs are not required.

RDW Maintenance



Pre-Batch Maintenance



Note:
salmth.pc resets the last eom_date. Thus, it must be run after the system indicator is extracted by pre_dwi_extract.ksh.

Post-Batch Maintenance

