Oracle® Retail Sales Audit

Operations Guide
Release 14.1
E58274-02

March 2015

ORACLE

Oracle® Retail Sales Audit Operations Guide, Release 14.1

E58274-02

Primary Author: Lydia Priyadarshini

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

Send Us Your Comments

Oracle Retail Sales Audit Operations Guide Release 14.1

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 new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. 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 www.oracle.com.

Preface

This *Oracle Retail Sales Audit Operations Guide* provides critical information about the processing and operating details of Oracle Sales Audit (ReSA), including the following:

- System configuration settings
- Technical architecture
- Functional integration dataflow across the enterprise
- Batch processing

Since ReSA is still closely integrated with the Oracle Retail Merchandising System (RMS) for data inputs, processes, and outputs, see the *Oracle Retail Merchandising System Operations Guide* for more information.

Audience

This guide is for:

- Systems administration and operations personnel
- Systems analysts
- Integrators and implementers
- Business analysts who need information about Merchandising System processes and interfaces

Documentation Accessibility

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

Access to Oracle Support

Oracle customers 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 Sales Audit documentation set:

- *Oracle Retail Merchandising System Release Notes*
- *Oracle Retail Sales Audit Installation Guide*
- *Oracle Retail Sales Audit User Guide*
- *Oracle Retail Sales Audit Operational Insights Reports User Guide*
- *Oracle Retail Merchandising Implementation Guide*
- *Oracle Retail Merchandising Security Guide*
- *Oracle Retail POS Suite 14.1/Merchandising Operations Management 14.1 Implementation Guide*
- *Oracle Retail Merchandising Batch Schedule*

- Oracle Retail Merchandising System documentation
- Oracle Retail Trade Management documentation

Customer Support

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

https://support.oracle.com

When contacting Customer Support, please provide the following:

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

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 14.1) or a later patch release (for example, 14.1.1). If you are installing the base release or additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch 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

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

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

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

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

Conventions

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."

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

1
Introduction

The purpose of the Oracle Retail Sales Audit (ReSA) is to accept transaction data from Point-Of-Sale (POS) and Order-Management-System (OMS) applications and move the data through a series of processes that culminates in *clean* data. Data that ReSA finds to be inaccurate is brought to the attention of the retailer's sales auditors who use the features of the sales audit system to correct the exceptions.

By using ReSA, retailers can quickly and accurately validate and audit transaction data before it is exported to other applications. ReSA uses several batch-processing modules to do the following:

- Import POS/OMS transaction data sent from the store to the ReSA database.
- Produce totals from user-defined totaling calculation rules that a user can review during the interactive audit.
- Validate transaction and total data with user-defined audit rules that generate errors whenever data does not meet the criteria. You can review these errors during the interactive audit.
- Create and export files in formats suitable for transfer to other applications.
- Update the ReSA database with adjustments received from external systems on previously exported data.

2

Technical Architecture

This chapter describes the overall software architecture for Oracle Retail Sales Audit and provides a high-level discussion of the general structure of the system, including the various layers of Java code.

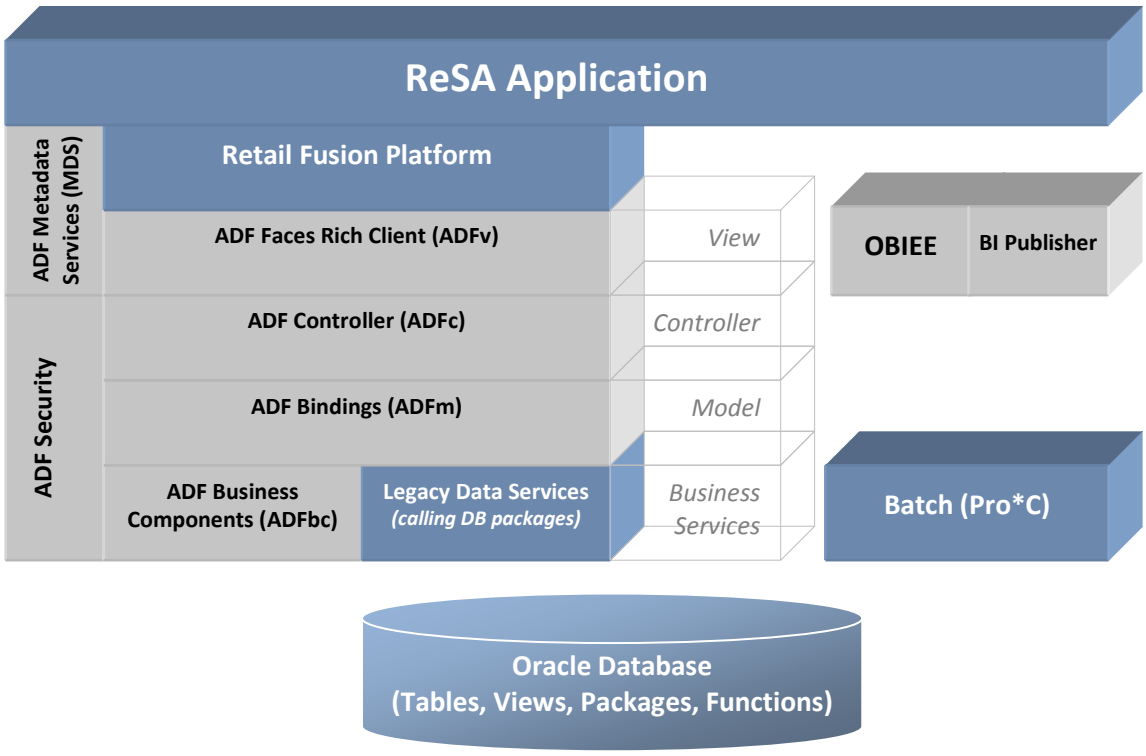
Overview

Retail applications are based on the Oracle Application Development Framework (ADF). The building blocks of ReSA are as follows:

- Oracle ADF application
- Pro*C batch jobs
- OBIEE and BI publisher for business intelligence reports.

The Oracle Database is integral to the application and the core business logic is implemented using PL/SQL packages/functions and used by the batch jobs, user interface and the business intelligence reports.

The following diagram shows the key components that make up the the architecture of ReSA.

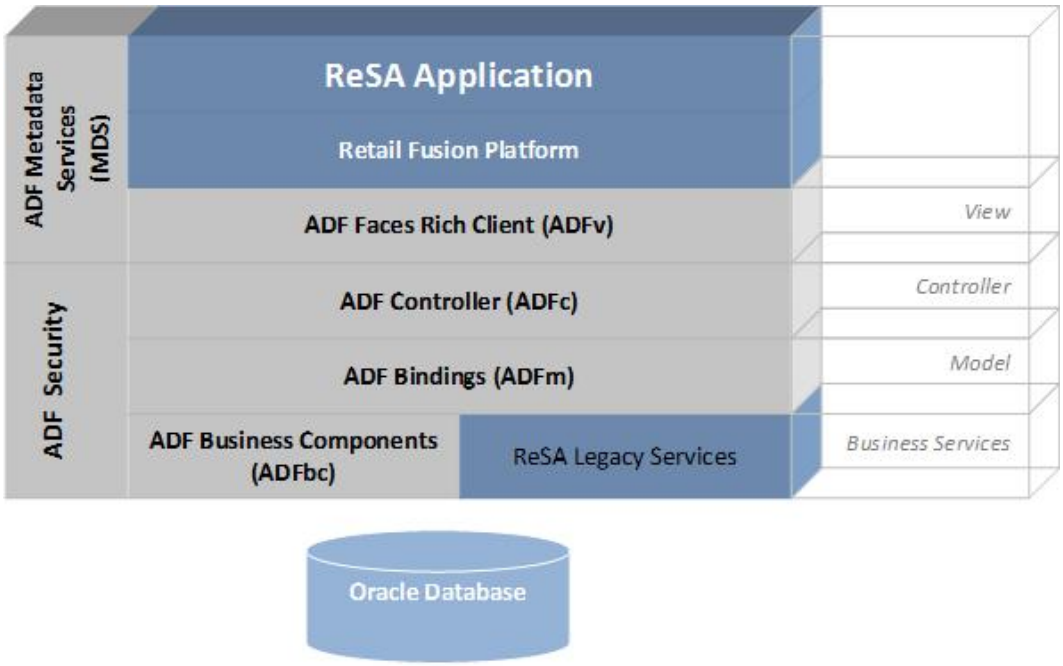


Batch Overview

The ReSA batch job implements the same Pro*C batch structure as the RMS. For more information on the RMS Pro*C architecture, see the *Oracle Retail Merchandising System Operation Guide*, Volume 3 - Back End Configuration and Operations.

Application Overview

The ReSA application is based on the Oracle Application Development Framework (ADF).



Oracle Application Development Framework (ADF)

Oracle Application Development Framework (ADF) supports organizations in building cutting-edge rich enterprise business applications that are able to be customized and personalized in all dimensions. Customizations are global changes, visible to all users, which are performed by an administrator. Personalizations are user-made changes that are only visible to the person making the change. ADF is based on the Java Enterprise Edition platform.

Model-View-Controller (MVC) Architectural Pattern

Applications built using ADF follow a Model-View-Controller (MVC) architectural pattern. The goal of the MVC pattern is to clearly separate the application's functionality into a set of cooperating components.

ADF provides a set of components that realize the goals of each part of the MVC pattern:

- Model is realized by the ADF Bindings Layer.
- Controller is realized by the ADF Controller Layer.
- View is realized by the ADF Faces Layer.
- ADF Business components and other back-end components that exist below the Model layer are called Business Services.

ADF Security

The ADF security layer provides the following:

- Provides standards based (Oracle Platform Security Services (OPSS)) security framework with default roles and permissions.
- Tools to generate file-based identity store (for both Oracle Internet Directory (OID) and Active Directory (AD)) based on the framework.
- Tools to migrate file-based security store into a database for Quality Assurance (QA) and production environments.
- Provides reference implementation for clients to manage the security based on their business needs.
- Provides OPSS-based batch security framework (RAF).
- Provides tools/documentation to implement centralized logout in Single Sign-On (SSO) (Oracle Access Management (OAM)) environments.

ADF View (ADFv)

The View layer provides the user interface to the application. The view layer uses HTML, rich Java components, or XML and its variations to render the user interface. JSF-based tag libraries are used to display the User Interface (UI).

ADF Controller (ADFc)

This is the steering part of the ADF. It determines the flow and control transfer. The Controller layer controls the application's flow. Web-based applications are composed of multiple web pages with dynamic content. The controller layer manages the flow between these pages. Different models can be used when building this later. The most prominent architecture for Java-based web applications relies on a servlet that acts as the controller. The Apache Jakarta Struts controller, an open source framework controller, is the de facto standard for Java-based web systems. Oracle ADF uses the struts controller to manage the flow of web applications.

ADF Business Components (ADFbc)

The business service layer manages the interaction with a data persistence layer. It provides services as data persistence, object/relational mapping, transaction management, and business logic execution.

Business Components easily map the database object and extend it with business logic, validation, and so on.

The idea behind Business Components is to abstract the data layer from the view layer. This is a key concept in the MVC pattern. Business Components expose the interface to the view layer by using an application module which contains View Object. Those view objects contain a specific usage of the data layer.

ADF Business Components implements the business service through the following set of cooperating components:

- Entity object – An entity object represents a row in a database table and simplifies modifying its data by handling all data manipulation language (DML) operations for you. It can encapsulate business logic for the row to ensure that your business rules are consistently enforced. You associate an entity object with others to reflect relationships in the underlying database schema to create a layer of business domain objects to reuse in multiple applications.
- View object – A view object represents a SQL query. You use the full power of the familiar SQL language to join, filter, sort, and aggregate data into exactly the shape required by the end-user task. This includes the ability to link a view object with others to create master-detail hierarchies of any complexity. When end users modify data in the user interface, view objects collaborate with entity objects to consistently validate and save the changes.
- Application module – An application module is the transactional component that UI clients use to work with application data. It defines a data model that you can update and top-level procedures and functions (called service methods) related to a logical unit of work related to an end-user task.

ADF Model (ADFm)

This is the component that acts as the connector between the view and business logic layers.

The Model layer connects the Business Services to the objects that use them in the other layers. Oracle ADF provides a Model layer implementation that sits on top of Business Services, providing a single interface that can be used to access any type of Business Services.

Developers get the same development experience when binding any type of Business Service layer implementation to the view and Controller layers. The Model layer in Oracle ADF served as the basis for JSR 227, A Standard Data binding & Data Access Facility for J2EE.

Oracle Metadata Services (MDS)

The ability of an application to adapt to changes is a necessity that needs to be considered in the application design and should drive the selection of the development platform and architecture. Flexible business applications must be able to adapt to organizational changes, different end user preferences, and changes in the supported business are required.

MDS is the customization and personalization framework integral to Oracle Fusion Middleware and a key differentiator of the Oracle development platform. MDS provides a repository for storing metadata for applications, such as customizations and persisted personalization files and configurations.

Retail applications allow the following through MDS:

- Personalization of saved searches through MDS.
- Implicit personalization of few ADF UI attributes.
- Customization of dashboards through Oracle Composer.

Retail Fusion Platform

The Retail Fusion Platform (commonly referred to as Platform) is a collection of common, reusable software components that serve as a foundation for building Oracle Retail's next generation ADF-based applications. The Platform imposes standards and patterns along with a consistent look and feel for Oracle Retail's ADF applications.

Data Access Patterns

Database interaction between the middle tier and database is done using the industry standard Java Database Connectivity Protocol (JDBC). JDBC facilitates the communication between a Java application and a relational database.

Database Access Using ADFbc

JDBC is engrained within Oracle ADF Business Components as the primary mechanism for its interaction between the middle tier and database. SQL is realized within ADF business components to facilitate create, read, update, and delete (CRUD) actions.

Connection Pooling

When the application *disconnects* a connection, the connection is saved into a pool instead of being actually disconnected. A standard connection pooling technique, this saved connection enables Retail Applications to reuse the existing connection from a pool. In other words, the application does not have to complete the connection process for each subsequent connection.

ReSA Database Services

ReSA Legacy services are implemented in PL/SQL packages. The invocation of these backend components are encapsulated in ADF business components, typically as Java wrappers that internally make use of JDBC to invoke the PL/SQL package calls.

To support ReSA legacy services implementing CRUD operations needing a dedicated connection, the standard application module connection pooling is enhanced to provide a dedicated connection till the end of the work flow. The special handling of reserving an AM connection is implemented only for scenarios that demand it.

Data Storage

The Oracle Database realizes the database tier in a Retail Application's architecture. It is the application's storage platform, containing the physical data (user and system) used throughout the application. The database tier is only intended to handle the storage and retrieval of information and is not involved in the manipulation or delivery of the data. This tier responds to queries; it does not initiate them.

Accessing Merchandising System Data in Real Time

The data that Retail Application utilizes is located in both application-specific tables and merchandising system (RMS, for example) tables. Because Retail Applications share the same schema as the merchandising system (RMS, for example), the application is able to interact with the merchandising system's data directly, in real time.

3

Administration and Configuration

This chapter is intended for administrators who support and monitor the running system.

This chapter discusses the Functional Security for Retail Applications and the components used to implement it. Functional security is based on Oracle Platform Security Services (OPSS). For more information on OPSS, see *Oracle Fusion Middleware Application Security Guide*.

The content in this chapter is not procedural, but is meant to provide descriptive overviews of the key system parameters.

This chapter consists of the following sections:

- Retail Roles
- Retail Role Hierarchy
- Default Security Reference Implementation
- Extend the Default Security Reference Implementation

Retail Roles

By default, users are not assigned to permissions directly as access is assigned to roles. Users, with roles assigned to them, group particular permissions required to accomplish a task. Instead of assigning individual permissions, roles match users with the permissions required to complete their particular task.

There are two main types of roles:

- Enterprise
- Application

The Identity Store contains enterprise roles that are available across applications. These are created as groups in Lightweight Directory Access Protocol (LDAP), making them available across applications.

Applicable Retail Applications security provides four types of roles:

- Abstract
- Job
- Duty
- Privilege

Applicable Retail Applications record abstract roles and jobs as enterprise roles. Duty and privilege roles are recorded as application roles.

Security Policy Stripe

Application roles are stored in the application specific policy store. These roles and role mappings are described in the `jazn-data.xml` file under the RESA policy stripe.

Abstract Roles

Abstract roles are associated with a user, irrespective of the user's job or job function. These are also roles that are not associated with a job or duty. These roles are normally

assigned by the system (based on user attributes), but can be provisioned to a user on request.

Naming Convention: All the Retail Abstract role names end with _ABSTRACT

Example: APPLICATION_ADMIN_ABSTRACT

Job Roles

Job roles are associated with the job of an employee. An employee, with this job, can have many job functions or job duties.

> **Note:** These roles are called Job roles as the role names closely map to the jobs commonly found in most organizations.

Naming Convention: All the Retail Job role names end with _JOB

Example: AUDITOR_JOB

Duty Roles

Job duties are tasks a person must do on a job. A person is hired into a job role. These are the responsibilities a person has for a job.

Duty roles are roles that are associated with a specific duty or a logical grouping of tasks. Generally, the list of duties for a job is a good indicator of what duty roles should be defined.

Duty roles should:

- Read as a job description at a job posting site
- Duties that we create should be self-contained and pluggable into any existing or new job or abstract role

Naming Convention: All the Retail duty role names end with _DUTY

Example: RESA_STOREDAY_MGMT_DUTY

Privilege Roles

Privilege is the logical collection of permissions. A privilege can be associated with any number of UI components. Privileges are expressed as application roles.

Naming Convention: All the Retail Privilege role names end with _PRIV

For example: SEARCH_STOREDAY_PRIV

Privilege roles carry security grants.

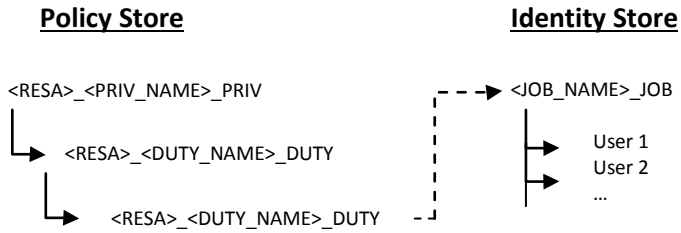
Example:

```
<grant>
                        <grantee>
                            <principals>
                                <principal>
                                    <name>SEARCH_STOREDAY_PRIV</name>

<class>oracle.security.jps.service.policystore.ApplicationRole</class>
                                </principal>
                            </principals>
                        </grantee>
                        <permissions>
                            <permission>

<class>oracle.adf.controller.security.TaskFlowPermission</class>
```

```
                                    <name>/WEB-
INF/oracle/retail/apps/resa/storeday/search/publicui/flow/SearchStoreDayFlow.xml#S
earchStoreDayFlow</name>
                                    <actions>view</actions>
                                </permission>
                            </permissions>
                        </grant>
```

Retail Role Hierarchy

Retail role hierarchies are structured to reflect the retail business process model.



Job roles inherit duty roles. For example, the Allocator Job role inherits the ALC_ALLOC_SYSTEM_OPTIONS_INQUIRY_DUTY roles.

```
<app-role>
  <name>RESA_STOREDAY_MGMT_DUTY</name>
  <class>oracle.security.jps.service.policystore.ApplicationRole</class>
  <members>
   <member>
     <class>oracle.security.jps.internal.core.principals.
               JpsXmlEnterpriseRoleImpl</class>
     <name>AUDITOR_JOB</name>
  </member>
 </members>
</app-role>
```

Duty roles inherit Privilege roles. Duty roles can inherit one or more other Duty roles. For example: RESA_STOREDAY_MGMT_DUTY inherits RESA_STOREDAY_INQUIRY_DUTY role.

```
<app-role>
  <name> RESA_STOREDAY_INQUIRY_DUTY</name>
  <class>oracle.security.jps.service.policystore.ApplicationRole</class>
  <members>
    <member>
     <class>oracle.security.jps.service.policystore.ApplicationRole</class>
     <name> RESA_STOREDAY_MGMT_DUTY</name>
    </member>
  </members>
</app-role>
```

For example: RESA_STOREDAY_INQUIRY_DUTY role inherits the SEARCH_STOREDAY_PRIV role

```
<app-role>
```

```
    <name>SEARCH_STOREDAY_PRIV</name>
       <class>oracle.security.jps.service.policystore.ApplicationRole</class>
        <description>A privilege for searching for store days.</description>
        <members>
           <member>
             <name>RESA_STOREDAY_INQUIRY_DUTY</name>

<class>oracle.security.jps.service.policystore.ApplicationRole</class>
           </member>
         </members>
  </app-role>
```

Default Security Reference Implementation

Retail applications ship with default security reference implementations. The source of truth for a default reference implementation is `jazn-data.xml`.

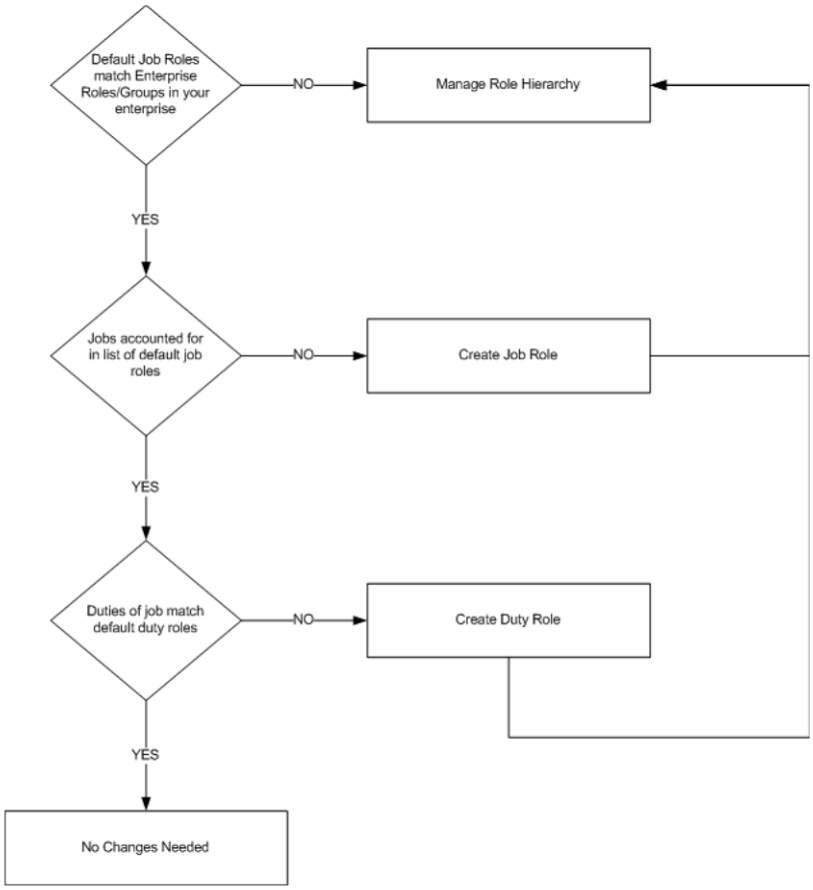
For more information on security, see the *Oracle Retail Merchandising Security Guide*.

Extending the Default Security Reference Implementation

The common decisions made to match your enterprise to the default security reference implementation include the following:

- Do the default job roles match the equivalent job roles in your enterprise?
- Do the jobs in your enterprise exist in the security reference implementation?
- Do the duties performed by the jobs in your enterprise match the duties in the security reference implementation?

> **Note:** Make sure that the policy store is loaded with the default security configuration. For more information, see the post-installation steps in the *Oracle Retail Sales Audit Installation Guide*.



Important: It is important when constructing a role hierarchy that circular dependencies are not introduced. Best practice is to leave the default security configuration in place and first incorporate your customized application roles in a test environment.

Using The Retail Application Security Role Manager

Retail Applications provides a way in which retailers can modify the default roles to map to their security groups through the Retail Application Security Role Manager (RASRM).

RASRM is installed along with the Retail Application. Users with proper security privileges to access RASRM can launch RASRM by clicking on a link from the Retail Application's global menu.

For more details about using RASRM, see the *Oracle Retail Merchandising Security Guide*.

Displaying External Application Contents in Non-SSO Environments

Retail Applications allow retailers to display content from external applications. These contents are typically business intelligence reports from a third-party application that are configured to display within the Retail Application's dashboard.

Some of these contents might be secured. You need to log in before the contents can be accessed and displayed.

In non-SSO environments, when you log out of the Retail Application, you may not be logged out of any secured content to which you had configured access. It is highly recommended that you only configure access to external content in a SSO-enabled environment where the application logout manages the logout from any other secured content that was previously accessed.

Managing Oracle Metadata Services (MDS)

Retail Applications are built using ADF, and one of the features within ADF is the Oracle Metadata Services (MDS) framework which provides a facility for retailers to customize the applications.

For more information about MDS, see *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework* (http://docs.oracle.com/cd/E16764_01/web.1111/b31974/customize.htm).

Purging the MDS Repository Before Patch Installation

A common problem when a patch is installed for Retail Application is that certain screens would fail to load or UI elements fail to display data properly.

The cause of this issue is commonly attributed to user personalizations on screen elements that are now removed in the patch.

For example, prior to patching the application, you may have saved search criterias on certain screens as a way to conveniently recall the desired search results whenever you use the application. Those saved search criterias are persisted by ADF in the MDS repository. If the patch involves the removal of one of the search criterias, applying the patch will cause the screens, that have those search criterias, to fail to load.

The MDS repository is configured in the WebLogic server where the Retail Application is deployed. The repository is database-based and is organized or subdivided into partitions. Retail Applications are deployed with their own partition within the server's MDS repository.

Prior to patch installation, it is recommended that retailers delete the MDS partition for the Retail Application in order to prevent runtime issues caused by stale user personalizations.

For ReSA, the MDS partition to be deleted is EarResaPortal.

To delete the partition, refer the Deleting a Metadata Partition Using Fusion Middleware Control section, in the *Oracle Fusion Middleware Administrator's Guide* (http://docs.oracle.com/cd/E23943_01/core.1111/e10105/repos.htm).

Note: After deleting the MDS partition, all user personalizations on the application will be lost. You need to reapply your personalizations on the application screens.

4

Oracle Retail Sales Audit Batch Processes and Designs

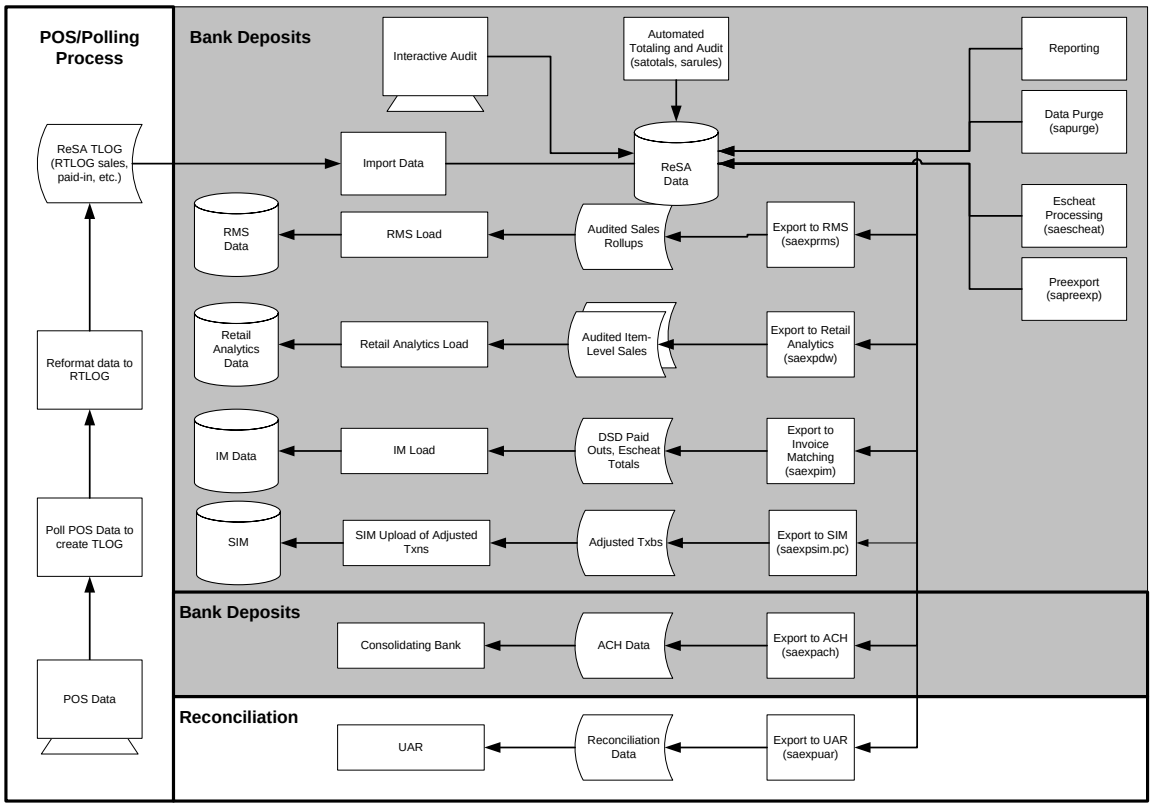
This chapter describes the batch processing modules that ReSA uses.

The term **store day** is used throughout this chapter. Store day describes all transactions that occur in one business day at one store or location. Because retailers need the ability to audit transactions on a store-by-store basis for a defined period of time, store day data is maintained separately beginning with the initial import of data from the POS/OMS system.

Oracle Retail Sales Audit Dataflow Diagram

The following diagram illustrates how data flows within ReSA and between ReSA and other applications.

Note: All integrations are not depicted in this diagram.



Oracle Retail Sales Audit Dataflow Diagram

Oracle Retail Sales Import Process

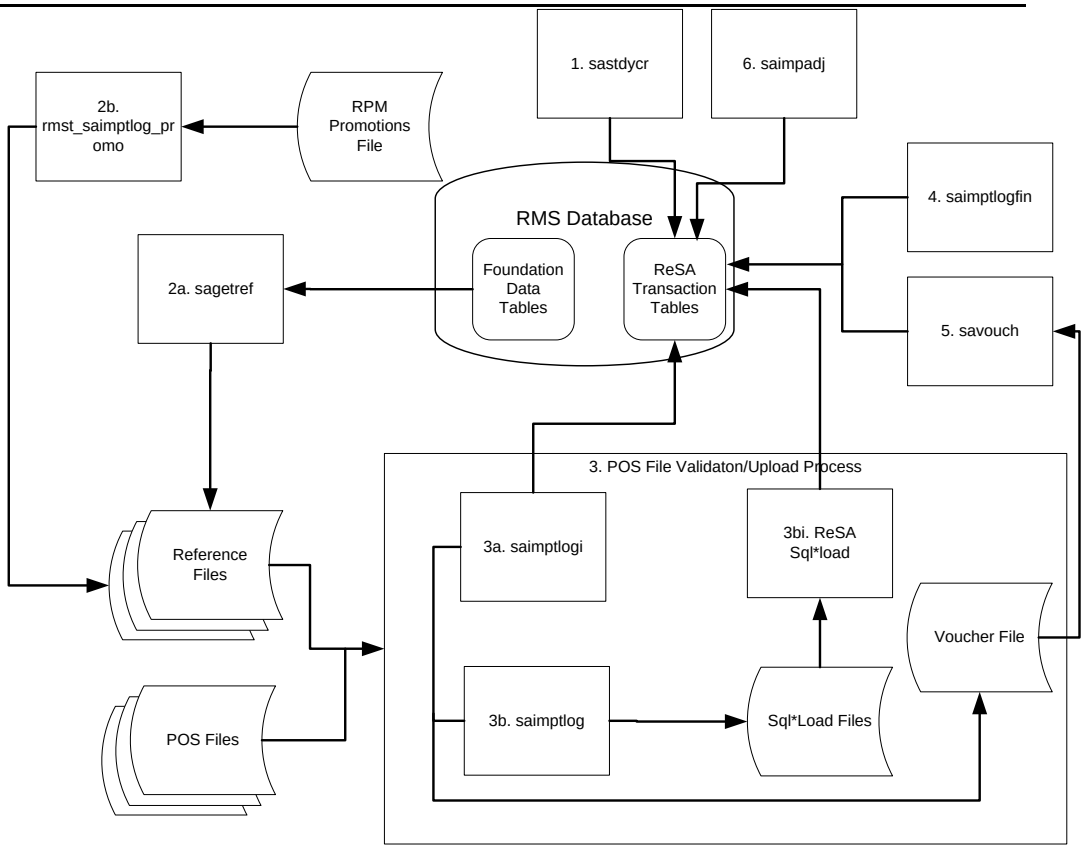
Importing data from the POS to ReSA is a multi-step process that involves several ReSA batch processes:

1. sastdycr.pc prepares the ReSA tables for data upload.
2. Reference File Creation involves two processes:
   a. sagetref.pc creates a number of reference files to be used for validation in the File Validation/Upload Process.
   - one of the reference data files created by sagetref contains code values from the RMS CODE_HEAD and CODE_DETAIL tables. If the codes that ReSA uses are customized during the implementation, a library must be recompiled. This is discussed in detail in the Design Assumptions section of the sagetref program level information below.
   - The way primary variants are set up in RMS affects the data collected by sagetref and used in the File Validation/Upload Process. This is discussed in detail in the Design Assumptions section of the sagetref program level information below.

   b. rmst_saimptlog_promo.ksh transforms a promotions file from Oracle Retail Price Management (RPM) to the ReSA reference file format
3. The POS File Validation/Upload Process can be executed one of two sub-processes:
   a. saimptlogi.c validates files and uploads their transactions into the ReSA tables. This includes (as necessary) creating errors for the auditors to address.
   b. saimptlog.c validates POS files and creates Sql*Loader Files. This includes (as necessary) creating errors for the auditors to address.
   - A Sql*Load process moves the transactions and errors into the ReSA tables.

   c. Both saimptlog and saimptlogi create a voucher file to be used in later processing.
4. saimptlogfin.pc executes a number of import cleanup processes.
5. savouch.pc processes voucher sales and redemptions.
6. saimpadj.pc imports adjustments.

Each of the processes related to the import process are discussed in more detail at the program level later in this chapter.



Oracle Retail Sales Import Process

POS File Validation/Upload Sub-Process – saimptlog vs. saimptlogi

saimptlogi.c and saimptlog.c perform the same business functions. Saimptlogi.c inserts directly into the database. Saimptlog uses SQL*Load to insert data. A retailer trickle polling or exporting a relatively small TLOG would be a good candidate to use saimptlogi.c. The detail discussion of these programs below contains more detail about the processing of these jobs.

Total Calculations and Rules

By providing additional values against which auditors can compare receipts, totaling is integral to the auditing process. Totaling also provides quick access to other numeric figures about the day's transactions.

Totaling in ReSA is dynamic. ReSA automatically totals transactions based on calculation definitions that the retailer's users create using the online Totals Calculation Definition Wizard. In addition, the retailer is able to define totals that come from the POS, but that ReSA does not calculate. Whenever users create new calculation definitions or edit existing ones, they become part of the automated totaling process the next time that satotals.pc runs.

Evaluating rules is also integral to the auditing process. Rules make the comparisons among data from various sources. These comparisons find data errors that could be the

result of either honest mistakes or fraud. Finding these mistakes during the auditing process prevents these errors from being passed on to other systems, (for example, a merchandising system, a data warehouse system, and so on).

Like totaling, rules in ReSA are dynamic. They are not predefined in the system—retailers have the ability to define them through the online Rules Calculation Definition Wizard.

Errors uncovered by these rules are available for review during the interactive audit. Like satotals.pc, after users modify existing rules or create new ones, they become part of the rules the next time that sarules.pc runs.

Totals when Transactions are Modified

If a retailer modifies transactions during the ReSA interactive audit process, the totaling and auditing processes run again to recalculate store day totals. The batch module sapreexp.pc tracks all changed totals for the store day since the last export by comparing the latest prioritized version of each total defined for export with the version that was previously sent to each system. The module writes the changes to revision tables that the export modules later recognize as ready for export.

Oracle Retail Sales Export Process

ReSA prepares data for export to applications after:

- Some or all of the transactions for the day are imported (depending upon the application receiving ReSA's export).
- Totals have run.
- Audit rules have run.
- Errors in transactions and totals relevant for the system receiving the associated data are eliminated or overridden. Depending upon the application, exported data consists of either transaction data or totals, or both. The process of exporting transaction data varies according to the unit of work selected in ReSA's system options. There are two units of work, transaction and store day. If the unit of work selection is transaction, ReSA exports its transactions to downstream applications, (for example, RMS, SIM, RA, etc.) as soon as they are free of errors. If the unit of work selection is store day, transactions are not exported until all errors for that store day are either overridden or corrected. The data export jobs, to the various downstream applications, can be run multiple times in a day.

Full Disclosure and Post-Export Changes

If a retailer modifies data during the interactive audit that was previously exported to RMS, ReSA export batch modules re-export the modified data in accordance with a process called *full disclosure*. Full disclosure means that any previously exported values (dollars, units, and so on) are fully backed out before the new value is sent.

For example: a transaction originally shows a sale of 12 items, and that transaction is exported. During the interactive audit, a retailer determines that the correct amount is 15 items, (where three are more than the original amount) and makes the change. ReSA then flags the corrected amount for export to the application.

The detail discussion of these programs below contains more detail about the processing of these jobs.

Batch Design Summary of ReSA Modules

The following list summarizes the ReSA batch modules that are involved with processing POS/OMS transaction data, audit totals and rules, exports to other applications, and modifications and adjustments.

Import Process Programs

- sastdycr.pc (Create Store Day for Expected Transactions)
- sagetref.pc (Get Reference Data for Sales Audit Import Processing)
- rmst_saimptlog_promo.ksh (Transform Promotion Reference File from RPM format to Sales Audit Import Processing File Format)
- saimptlog.c/saimptlogi.c (Import of Unaudited Transaction data from POS to ReSA)
- saimptloglogtdup_upd (Processing to Allow Re-Upload of Deleted Transactions)
- saimptlogfin.pc (Complete Transaction Import Processing)
- savouch.pc (Sales Audit Voucher Upload)
- saimpadj.pc (Import Total Value Adjustments From External Systems to ReSA)

Totals/Rules Programs

- satotals.pc (Calculate Totals based on Client Defined Rules)
- sarules.pc (Evaluate Transactions and Totals based on Client Defined Rules)

Export Programs

- sapreexp.pc (Prevent Duplicate Export of Total Values from ReSA)
- saexprms.pc (Export of POS transactions from ReSA to RMS)
- saordinvexp.pc (Export Inventory Reservation/Release for In Store Customer Order and Layaway Transactions from ReSA)
- saexpdw.pc (Export from ReSA to Oracle Retail Analytics)
- saexpsim.pc (Export of Revised Sale/Return Transactions from ReSA to SIM)
- saexpim.pc (Export DSD and Escheatment from ReSA to Invoice Matching)
- saexpgl.pc (Post User Defined Totals from ReSA to General Ledger)
- ang_saplgen.ksh (Extract of POS Transactions by Store/Date from ReSA for Web Search)
- saescheat.pc (Download of Escheated Vouchers from ReSA for Payment)
- saescheat_nextesn.pc (Generate Next Sequence for Escheatment Processing)
- saexpach.pc (Download from ReSA to Account Clearing House (ACH) System)
- saexpuar.pc (Export to Universal Account Reconciliation System from ReSA)

Other programs

- saprepost.pc (Pre/Post Helper Processes for ReSA Batch Programs)
- sapurge.pc (Purge Aged Store/Day Transaction, Total Value, and Error Data from ReSA)

sastdycr (Create Store Day for Expected Transactions)

Module Name	sastdycr.pc
Description	Create Store Day for Expected Transactions
Functional Area	Oracle Retail Sales Audit
Module Type	Business Processing
Module Technology	ProC
Integration Catalog ID	RSA15

Design Overview

The sastdycr batch program will create store/day, import log, and export log records. This program should run prior to uploading the sales data from POS/OMS for a given store/day. Store/days will be created for any open store expecting sales.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Daily – In the date set phase.
Scheduling Considerations	It should run before the DTESYS batch program and before the next store/day's transactions are received.
Pre-Processing	NA
Post-Processing	dtesys
Threading Scheme	NA

Restart/Recovery

The logical unit of work in this program is store. Records are committed to the database when the commit counter is reached. The commit counter is defined by the value of INCREMENT_BY on the ALL_SEQUENCE table for the sequence SA_STORE_DAY_SEQ_NO_SEQUENCE.

Key Tables Affected

Table	Select	Insert	Update	Delete
ALL_SEQUENCES	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No
STORE	Yes	No	No	No
SA_STORE_DAY	Yes	Yes	No	No

Table	Select	Insert	Update	Delete
COMPANY_CLOSED	Yes	No	No	No
COMPANY_CLOSED_EXCEP	Yes	No	No	No
LOCATION_CLOSED	Yes	No	No	No
PERIOD	Yes	No	No	No
SA_STORE_DATA	Yes	No	No	No
SA_IMPORT_LOG	No	Yes	No	No
SA_EXPORT_LOG	No	Yes	No	No
SA_FLASH_SALES	No	Yes	No	No

Integration Contract

Integration Type	NA
File Name	NA
Integration Contract	NA

Design Assumptions

NA

sagetref (Get Reference Data for Sales Audit Import Processing)

Module Name	sagetref.pc
Description	Get Reference Data for Sales Audit Import Processing
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA00

Design Overview

This program will fetch all reference information needed by SAIMPTLOG.PC for validation purposes and write this information out to various output files. The following files are produced:

- Items - contains a listing of all items in the system.
- Wastage - contains information about all items that have wastage associated with them.
- Reference Items - contains reference items, or below transaction-level items.
- Primary Variant - contains primary variant information.
- Variable Weight UPC - contains all variable weight Universal Product Code (UPC) definitions in the system.
- Store/Days - contains all of the valid store/day combinations in the system.
- Codes & Code Types - contains all code types and codes used in field level validation.
- Error Codes & Descriptions - contains all error codes, error descriptions, and systems affected by the error.
- Store POS Mappings
- Tender Types
- Merchants
- Partners
- Suppliers
- ReSA Employees
- Banners
- Currency Codes
- Promotions (from RPM)
- Physical Warehouses
- Inventory Statuses

These files will be used by the automated audit to validate information without repeatedly hitting the database.

When running sagetref.pc, retailers can either create and specify the output files, or create only the output that they desire. For example, a retailer interested in only creating a more recent employeefile would simply place a hyphen (-) in place of all the other parameters, but still specify an employeefile name. This technique can be applied to as many or as few of the parameters as retailers wish. Note, however, that the item-related files (itemfile, refitemfile, wastefile, and primvariantfile) contain significant interdependence. Thus, item files must all be created or not created together.

In the list of reference data files files above, standard UOM is part of the itemfile. To obtain the value, ReSA converts the selling Unit of Measure (UOM) to the standard UOM during batch processing. This conversion enables ReSA to later export the standard UOM to the systems that require its use.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Daily – Anytime – Sales Audit is a 24/7 system.
Scheduling Considerations	This module should be executed in the earliest phase, before the first import of RTLOGs into ReSA.
Pre-Processing	Sastrdaycr.pc
Post-Processing	Saimptlog.pc or saimptlogi.pc
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table	Select	Insert	Update	Delete
ITEM_MASTER	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
VAR_UPC_EAN	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_STORE	Yes	No	No	No
SA_IMPORT_LOG	Yes	No	No	No
CURRENCIES	Yes	No	No	No
ADDR	Yes	No	No	No
CODE_DETAIL	Yes	No	No	No
SA_ERROR_CODES	Yes	No	No	No
SA_STORE_POS	Yes	No	No	No
POS_TENDER_TYPE_HEAD	Yes	No	No	No
NON_MERCH_CODE_HEAD	Yes	No	No	No
PARTNER	Yes	No	No	No

Table	Select	Insert	Update	Delete
SUPS	Yes	No	No	No
SA_STORE_EMP	Yes	No	No	No
STORE	Yes	No	No	No
BANNER	Yes	No	No	No
CHANNELS	Yes	No	No	No
CLASS	Yes	No	No	No
VAT_CODES	Yes	No	No	No
RPM_PROMO_V	Yes	No	No	No
RPM_PROMO_COMP_V	Yes	No	No	No
WH	Yes	No	No	No
INV_STATUS_CODES	Yes	No	No	No
SA_STORE_DATA	Yes	No	No	No

Integration Contract

Integration Type	Download from RMS
File Name	Determined by runtime parameters
Integration Contract	IntCon000113 (itemfile) IntCon000114 (wastefile) IntCon000115 (refitemfile) IntCon000116 (primvariantfile) IntCon000117 (varupcfile) IntCon000118 (storedayfile) IntCon000119 (promfile) IntCon000120 (codesfile) IntCon000121 (errorfile) IntCon000122 (storeposfile) IntCon000123 (tendertypefile) IntCon000124 (merchcodesfile) IntCon000125 (partnerfile) IntCon000126 (supplierfile) IntCon000127 (employeefile) IntCon000128 (bannerfile) IntCon000129 (promfile) IntCon000130 (whfile) IntCon000131 (invstatusfile)

File Name: Item File

The Item File file name (Itemfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Item	Char(25)		Item number
	Dept	Number(4)		Department ID
	Class	Number(4)		Class id
	Subclass	Number(4)		Subclass ID
	Standard UOM	Char(4)		Standard Unit of Measure
	Catchweight Ind	Char(1)		Catch weight indicator
	Class vat Ind	Char(1)		Class Vat Ind

File Name: Waste Data File

The Waste Data File file name (wastefile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Item	Char(25)		Item number
	Waste type	Char(6)		Waste type
	Waste pct	Number(12,4)		Waste pct

File Name: Reference Item Data

The Reference Item Data file name (ref_itemfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Ref Item	Char(25)		Reference Item number
	Item	Char(25)		Item number

File Name: Primary Variant Data File

The Primary Variant Data File file name (prim_variantfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Location	Number(10)		Location number
	Item	Char(25)		Item number
	Prim Variant	Char(25)		Primary variant

File Name: Variable Weight UPC Definition File

The Variable Weight UPC Definition File file name (varupcfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Format Id	Char(1)		Format id
	Format desc	Char(20)		Format description
	Prefix length	Number(1)		Pefix Length
	Begin item digit	Number(2)		Item digit begin
	Begin var digit	Number(2)		Var digit begin
	Check digit	Number(2)		Check digit
	Default prefix	Number(1)		Default prefix
	Prefix	Number(1)		Prefix

File Name: Valid Store/Day Combination File

The Valid Store/Day Combination File file name (storedayfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Store	Number(10)		Store number
	Business date	Char(8)		Buisness date in YYYYMMDD format
	Store day seq no	Number(20)		Store day sequence number
	Day	Number(3)		Day
	Tran no generated	Char(6)		Generated transaction number
	POS data expected	Char(1)		If system_code is POS, then Y; otherwise N
	Currency rtl dec	Number(1)		Currency rtl dec
	Currency code	Char(3)		Currency code
	Country id	Char(3)		Country ID
	Vat Include Ind	Char(1)		Vat Include Indicator

File Name: Codes File

The Codes File file name (codesfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Code type	Char(4)		Code type
	Code	Char(6)		Code ID
	Code seq	Number(4)		Code sequence

File Name: Error Information File

The Error Information File file name (errorfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Error code	Char(25)		Error code
	System Code	Char(6)		System Code
	Error desc	Char(255)		Error description
	Rec solution	Char(255)		Error rectify solution

File Name: Store POS Mapping File

The Store POS Mapping File file name (storeposfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Store	Number(10)		Store
	POS Type	Char(6)		Point Of Sale type
	Start Tran No.	Number(10)		Start transaction number
	End Tran No.	Number(10)		End transaction number

File Name: Tender Type Mapping File

The Tender Type Mapping File file name (tendertypefile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Group	Char(6)		Tender type Group
	Id	Number(6)		Tender type ID
	Desc	Char(120)		Tender type description

File Name: Merchant Code Mapping File

The Merchant Code Mapping File file name (merchcodesfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Non Merch Code	Char(6)		Non-merchant code

File Name: Partner Mapping File

The Partner Mapping File file name (partnerfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Partner Type	Char(6)		Partner Type
	Partner Id	Char(10)		Partner ID

File Name: Supplier Mapping File

The Supplier Mapping File file name (supplierfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Supplier	Number(10)		Supplier ID
	Sup status	Char(1)		Supplier status

File Name: Employee Mapping File

The Employee Mapping File file name (employeefile) is not fixed; it is determined by a runtime parameter.

Note: The data in the employee file is not used anymore with the removal of store user audit and the deprecation of system option, AUTO_VALIDATE_TRAN_EMPLOYEE_ID.

Record Name	Field Name	Field Type	Default Value	Description
	Store	Number(10)		Store ID
	POS Id	Char(10)		Point Of Sale ID
	Emp Id	Char(10)		Employee ID

File Name: Banner Information File

The Banner Information File file name (bannerfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Store	Number(10)		Store ID
	Banner data	Number(4)		Banner ID

File Name: Currency Information File

The Currency Information File file name (currencyfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description

	Currency Code	Char(1)	Currency Code

File Name: Promotion Information File

The Promotion Information File file name (promfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Promotion	Number(10)		Promotion ID
	Component	Number(10)		Component ID

File Name: Physical Warehouse Information File

The Physical Warehouse Information File filename (whfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Warehouse	Number(10)		Warehouse ID

File Name: Inventory Status Information File

The Inventory Status Information File file name (invstatusfile) is not fixed; it is determined by a runtime parameter.

Record Name	Field Name	Field Type	Default Value	Description
	Inventory Status	Char(10)		Inventory Status

Design Assumptions

Making Changes in the CODE_DETAIL Table

After making changes in the code_detail table for code_types that ReSA uses, the library programs must be recompiled. Follow these steps:

1. Navigate to the $l directory and recompile libresa.a and install:
   ```
   make -f retek.mk resa
   make -f retek.mk install
   ```
2. Navigate to the $c directory and recompile the next libraries:
   ```
   make -f mts.mk resa-libchange
   make -f mts.mk resa
   ```
   a. Recompile the appropriate library depending upon which of the following products is being used:
      - resa-rms
      - resa-rdw
      - resa-ach
      - resa-uar

- resa-im

```
make -f mts.mk ( name of library )
```

b. make -f mts.mk resa-install

Primary Variant Relationships

Depending upon a retailer's system parameters, the retailer designates the primary variant during item setup (through the front-end) for several reasons. One of the reasons is that, in some cases, an item may be identified at the POS by the item parent, but the item parent may have several variants.

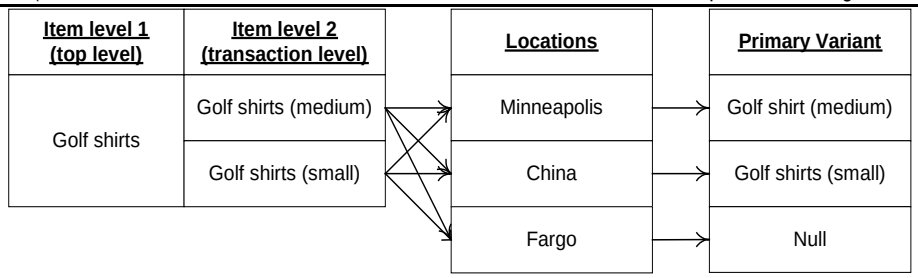
The primary variant is established through a form at the item location level. The retailer designates which variant item is the primary variant for the current transaction level item. For more information about the new item structure in RMS, see the *Oracle Retail Merchandising System User Guide*.

In the example shown in the diagram below, the retailer has established their transaction level as an Item Level 2. Note that the level of the primary variant is Item Level 1, and Item Level 3 is the sub-transaction level (the refitem).

The retailer set up golf shirts in the merchandising system as its Item Level 1 above the transaction level. The retailer set up two items at level 2 (the transaction level) based on size (small and medium). Note that the retailer assigned the level 2 items to all of the available locations (Minneapolis, China, and Fargo). The retailer also designated a primary variant for a single location – a medium golf shirt, in the case of Minneapolis, and a small golf shirt, in the case of China. The retailer failed to designate a primary variant for Fargo.

The primary variant affects ReSA in the following way. Sometimes a POS system does not provide ReSA with item level 2 (transaction item) data. For example, assume that the POS system in Minneapolis sold 10 medium golf shirts and 10 small golf shirts but only informed ReSA that 20 golf shirts were sold. 20 golf shirts presents a problem for ReSA because it can only interpret items at item level 2 (the transaction level). Thus, because medium golf shirts was the chosen primary variant for Minneapolis, the SAGETREF.PC module automatically transforms the 20 golf shirts into 20 medium golf shirts. If the same type of POS system in China informed ReSA of 20 golf shirts (instead of the 10 medium and 10 small that were sold), the sagetref.pc module would transform the 20 golf shirts sold in China into 20 small golf shirts. As the table shows, small golf shirts was the chosen primary variant for the China location. ReSA then goes on to export the data at the item 2 level (the transaction level) to, for example, a merchandising system, a data warehouse, and so on.

> **Note:** Depending upon system parameters, if a retailer fails to set up the primary variant for a location, an invalid item error is generated during batch processing. In the example below, if the POS system in Fargo sold 10 medium golf shirts and 10 small golf shirts, but only informed ReSA that 20 golf shirts were sold, the sagetref.pc module would not have a way to transform those 20 golf shirts to the transaction level. Because ReSA can only interpret items above the transaction level in conjunction with a primary variant, the invalid item error would occur during batch processing.



Primary Variant Relationships

rmst_saimptlog_promo (Transform Promotion Reference File from RPM Format to Sales Audit Import Processing File Format)

Module Name	rmst_saimptlog_promo.ksh
Description	Transform Promotion Reference File from RPM format to Sales Audit Import Processing file format
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	Ksh
Integration Catalog ID	RSA13

Design Overview

This script converts the prmdtldm.txt file (which is extracted by the prmdtlex.ksh Oracle Retail Extract, Transform, and Load (RETL) script from the RPM application) into the file format expected by the ReSA module saimptlog.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	NA
Pre-Processing	prmdtlex.ksh RETL script from the RPM application
Post-Processing	saimptlog.pc
Threading Scheme	NA

Restart/Recovery

This is a standard RETL script. No restart/recovery is used.

Key Tables Affected

NA

Integration Contract

Integration Type	Upload to ReSA
File Name	Determined by runtime parameter.
Integration Contract	IntCon000006 (prmdtldm.txt input) IntCon000119 (promfile output) rmst_saimptlog_promo.schema

Design Assumptions

NA

saimptlog/saimptlogi (Import of Unaudited Transaction Data from POS to ReSA)

Module Name	saimptlog.c saimptlogi.c
Description	Import of Unaudited Transaction data from POS to ReSA
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA11a RSA11b

Design Overview

Importing POS and Order Management System (OMS) data to ReSA is a 5 or 6 step process depending on whether saimptlogi or saimptlog is used. Saimptlog produces SQL*Loader files, while saimptlogi does inserts directly into the database. Saimptlogi is meant for use in a trickle feed environment.

To import POS and OMS data, perform the following:

1. SAGETREF must be run to generate the current reference files:
   - Items
   - Wastage
   - Sub-transaction level items
   - Primary variant relationships
   - Variable weight PLU
   - Store business day
   - Code types
   - Error codes
   - Store POS
   - Tender type
   - Merchant code types
   - Partner vendors
   - Supplier vendors
   - Employee ids
   - Banner ids
   - Currency File
   - Warehouse File
   - Inventory Status File

These files are all used as input to SAIMPTLOG and SAIMPTLOGI. Because SAIMPTLOG and SAIMPTLOGI can be threaded, this boosts performance by limiting interaction with the database.

2. Either SAIMPTLOG or SAIMPTLOGI must be run against each file. The files are the transaction log files in Oracle Retail compatible format called RTLOG. The retailer is responsible for converting its transaction logs to RTLOGs. Both SAIMPTLOG and SAIMPTLOGI create a write lock for a store/day combination on ReSA tables and then set the data_status to loading until SAIMPTLOGFIN is executed. SAIMPTLOG generates distinct SQL*Loader files for that store/day for the sa_tran_head, sa_tran_item, sa_tran_disc, sa_tran_igtax (item Level Tax not VAT), sa_tran_payment (Payment details), sa_tran_tax, sa_tran_tender, sa_error, sa_customer, sa_cust_attrib, and sa_missing_tran tables, whereas SAIMPTLOGI inserts data to the database directly. Both produce an Oracle Retail formatted voucher file for processing.
3. SQL*Loader is executed to load the transaction tables from the files created by SAIMPTLOG. The store/day SQL*Loader files can be concatenated into a single file per table to optimize load times. Alternatively, multiple SQL*Loader files can be used as input to SQL*Loader. SQL*Loader may not be run in parallel with itself when loading a table. Header data (primary keys) must be loaded before ancillary data (foreign keys). This means that the sa_tran_head table must be loaded first, sa_tran_item before sa_tran_disc, and sa_customer before sa_cust_attrib. The remaining tables may be loaded in parallel.
4. SAVOUCH is executed to load each of the voucher files in Oracle Retail standard formatted. SAVOUCH may not be multi-threaded.
5. SAIMPTLOGFIN is executed to populate the sa_balance_group table, cancel post voided transactions and vouchers, validate missing transactions, and mark the import as either partially or fully complete loaded. SAIMPTLOGFIN may not be multi-threaded.

Note: This design covers only Steps 2 and 3.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	saimptlog.c or saimptlogi.c should run after sagetref.pc to get the reference files.
Pre-Processing	saprepost saimptlog pre – change constraints on ReSA tables or saprepost saimptlogi pre – change constraints on ReSA tables.
Post-Processing	saprepost saimptlog post – change back constraints on ReSA tables or saprepost saimptlogi post – change back constraints on ReSA tables. sqlldr – use sql loader to load data into ReSA tables (for saimptlog only).
Threading Scheme	saimptlog and saimptlogi may be threaded as long as the parallel executions do not include the same store/day.

Restart/Recovery

NA

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_ROUNDING_RULE_HEAD	Yes	No	No	No
SA_ROUNDING_RULE_DETAIL	Yes	No	No	No
SA_STORE_DAY	Yes	No	Yes	No
SA_TRAN_HEAD	No	Yes	No	No
SA_CUSTOMER	No	Yes	No	No
SA_CUST_ATTRIB	No	Yes	No	No
SA_TRAN_ITEM	No	Yes	No	No
SA_TRAN_IGTAX	No	Yes	No	No
SA_TRAN_DISC	No	Yes	No	No
SA_TRAN_TAX	No	Yes	No	No
SA_TRAN_TENDER	No	Yes	No	No
SA_TRAN_PAYMENT	No	Yes	No	No
SA_ERROR	No	Yes	No	No
SA_MISSING_TRAN	No	Yes	No	No

Table	Select	Insert	Update	Delete
ALL_SEQUENCES	Yes	No	No	No

Integration Contract

Integration Type	Upload to ReSA
File Name	Determined by runtime parameter
Integration Contract	Inputs from sagetref.pc: IntCon000113 (itemfile) IntCon000114 (wastefile) IntCon000115 (refitemfile) IntCon000116 (primvariantfile) IntCon000117 (varupcfile) IntCon000118 (storedayfile) IntCon000119 (promfile) IntCon000120 (codesfile) IntCon000121 (errorfile) IntCon000122 (storeposfile) IntCon000123 (tendertypefile) IntCon000124 (merchcodesfile) IntCon000125 (partnerfile) IntCon000126 (supplierfile) IntCon000127 (employeefile) IntCon000128 (bannerfile) IntCon000129 (promfile) IntCon000130 (whfile) IntCon000131 (invstatusfile)
	Inputs from POS: IntCon000048 (RTLOG)
	Outputs (if using saimptlog SQL Loader Option – note that saimptlogi inserts directly into ReSA tables and does not create these output files) IntCon000160 (SAVO) IntCon000161 (satdisc.ctl) IntCon000162 (saigtax.ctl) IntCon000163 (sacust.ctl) IntCon000164 (sathead.ctl) IntCon000165 (satitem.ctl) IntCon000166 (sattend.ctl) IntCon000167 (satypmt.ctl)

	IntCon000168 (samisstr.ctl) IntCon000169 (sattax.ctl) IntCon000170 (sacustatt.ctl) IntCon000171 (saerror.ctl)

The input files for this program are reference files generated by sagetref.pc and RTLOGs. For the input file specificatiosn, refer to the details for the sagetref.pc program.

Output File Layout

File Name: SAVO (Sales Audit Voucher File)

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File Type Record Descriptor	Char(5)	FHEAD	File type Record descriptor
	SA File Line No	Char(10)		Sales Audit File Line number
	Translator Id	Char(5)	SAVO	Identifies transaction type
	Sys Date	Char(14)		System date in YYYYMMDDHHMMSS format
	Is business date	Char(8)		Business date in YYYYMMDD format
FDETL	Record Descriptor	Char(5)	FDETL	File Type Record descriptor
	SA File Line No	Number(10)		Sales Audit File Line number
	Voucher seq Number	Number(20)		Unique identifier for an entry to sa_voucher table
	Voucher No	Char(25)		Voucher Number
	Voucher Type	Number(6)		Voucher Type
	Assigned Business Date	Char(8)		Business date in YYYYMMDD format
	Assigned Store	Number(10)		Store to which the voucher is assigned
	Issuing Date	Char(8)		Date this document was issued
	Issuing store	Number(10)		Store this document was issued from
	Issuing POS Register	Char(5)		Issuing Point Of Sale register
	Issuing Cashier	Char(10)		Issuing cashier
	Issued Tran Seq No.	Number(20)		Transaction sequence number

Record Name	Field Name	Field Type	Default Value	Description
	Issued item seq number	Number(4)		Will hold the item sequence of the item when the voucher is sold as an item (gift voucher)
	Issued Tender Seq No.	Number(4)		Tender sequence number
	Issued Amount	Number(20)		Issued Amount * 10000 (4 implied digits)
	Issued Cust Name	Char(120)		Issued customer name
	Issued Customer Addr1	Char(240)		Issued customer addr1
	Issued Customer Addr2	Char(240)		Issued customer addr 2
	Issued Customer City	Char(120)		City of the customer, the voucher is issued
	Issued Customer State	Char(3)		State of the customer
	Issued Customer Postal Code	Char(30)		Postal address of the customer
	Issued Customer Country	Char(3)		Country of the customer the voucher was issued
	Recipient Name	Char(120)		Name of the intended recipient
	Recipient State	Char(3)		The state of the intended recipient
	Recipient Country	Char(3)		The country of the intended recipient
	Redemption Date	Char(8)		Date the voucher was redeemed
	Redemption Store	Number(10)		Store, the voucher was redeemed at
	Redemption Register	Char(5)		Register, the document was redeemed at
	Redemption cashier	Char(10)		Cashier redeeming the voucher
	Redemption tran seq number	Number(20)		Transaction number when the document was redeemed
	Redemption Tender seq number	Number(4)		This column will hold the tender sequence of the tender within the transaction when a voucher is redeemed as tender

Record Name	Field Name	Field Type	Default Value	Description
	Redemption Amount	Number(20)		Amount the document was redeemed for*10000 (4 implied decimal places)
	Expiry Date	Char(8)		Expiry date
	Status	Char(1)		Indicator showing the document's status, issued or redeemed. Valid values = I – Issued, R – Redeemed
	Comments	Char(2000)		Comments
FTAIL	Record Descriptor	Char(5)	FTAIL	File Type Record descriptor
	SA File Line No.	Number(10)		Sales Audit File Line Number
	#lines	Number(10)		Total number of transaction lines in the file (not including FHEAD and FTAIL)

Control Files

File Name: Satdisc.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_DISC	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ITEM_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	DISCOUNT_SEQ_NO	INTEGER EXTERNAL	4	25:28	
	RMS_PROMO_TYPE	CHAR	6	29:34	
	PROMOTION	INTEGER EXTERNAL	10	35:44	
	DISC_TYPE	CHAR	6	45:50	
	COUPON_NO	CHAR	16	51:66	
	COUPON_REF_NO	CHAR	16	67:82	
	QTY	DECIMAL EXTERNAL	14	83:96	
	UNIT_DISCOUNT_AMT	DECIMAL EXTERNAL	21	97:117	
	STANDARD_QTY	DECIMAL EXTERNAL	14	118:131	
	STANDARD_UNIT_DISC_AMT	DECIMAL EXTERNAL	21	132:152	
	REF_NO13	CHAR	30	153:182	

Table Name	Column Name	Field Type	Field Width	Position	Description
	REF_NO14	CHAR	30	183:212	
	REF_NO15	CHAR	30	213:242	
	REF_NO16	CHAR	30	243:272	
	ERROR_IND	CHAR	1	273:273	
	CATCHWEIGHT_IND	CHAR	1	274:274	
	UOM_QUANTITY	INTEGER EXTERNAL	12	275:286	
	PROMO_COMP	INTEGER EXTERNAL	10	287:296	
	STORE	INTEGER EXTERNAL	10	297:306	
	DAY	INTEGER EXTERNAL	3	307:309	

File Name: Saigtax.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_IGTAX	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ITEM_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	IGTAX_SEQ_NO	INTEGER EXTERNAL	4	25:28	
	TAX_AUTHORITY	INTEGER EXTERNAL	10	29:38	
	IGTAX_CODE	CHAR	6	39:44	
	IGTAX_RATE	DECIMAL EXTERNAL	11	45:65	
	TOTAL_IGTAX_AMT	DECIMAL EXTERNAL	22	66:87	
	STANDARD_QTY	DECIMAL EXTERNAL	14	88:101	
	STANDARD_UNIT_IGTAX_AMT	DECIMAL EXTERNAL	21	102:122	
	ERROR_IND	CHAR	1	123:123	
	REF_NO_21	CHAR	30	124:153	
	REF_NO_22	CHAR	30	154:183	
	REF_NO_23	CHAR	30	184:213	

Table Name	Column Name	Field Type	Field Width	Position	Description
	REF_NO_24	CHAR	30	214:243	
	STORE	INTEGER EXTERNAL	10	244:253	
	DAY	INTEGER EXTERNAL	3	254:256	

File Name: Sacust.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_CUSTOMER	TRAN_SEQ_NO	INTEGER EXTERNAL DATE	20	1 :20	
	CUST_ID	CHAR	16	21 :36	
	CUST_ID_TYPE	CHAR	6	37 :42	
	NAME	CHAR	240	43 :162	
	ADDR1	CHAR	240	163:402	
	ADDR2	CHAR	240	403:642	
	CITY	CHAR	240	643:762	
	STATE	CHAR	3	763:765	
	POSTAL_CODE	CHAR	30	766:795	
	COUNTRY	CHAR	3	796:798	
	HOME_PHONE	CHAR	20	799:818	
	WORK_PHONE	CHAR	20	819:838	
	E_MAIL	CHAR	100	839:938	
	BIRTHDATE	DATE	8	939:946	Format is YYYYMMDD
	STORE	INTEGER EXTERNAL	10	947:956	
	DAY	INTEGER EXTERNAL	3	957:959	

File Name: Sathead.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_HEAD	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	

Table Name	Column Name	Field Type	Field Width	Position	Description
	REV_NO	INTEGER EXTERNAL	3	21:23	
	STORE_DAY_SEQ_NO	INTEGER EXTERNAL	20	24:43	
	TRAN_DATETIME	DATE	14	44:57	Format is YYYYMMDDHH24MISS
	REGISTER	CHAR	5	58:62	
	TRAN_NO	INTEGER EXTERNAL	10	63:72	
	CASHIER	CHAR	10	73:82	
	SALESPERSON	CHAR	10	83:92	
	TRAN_TYPE	CHAR	6	93:98	
	SUB_TRAN_TYPE	CHAR	6	99:104	
	ORIG_TRAN_NO	INTEGER EXTERNAL	10	105:114	
	ORIG_REG_NO	CHAR	5	115:119	
	REF_NO1	CHAR	30	120:149	
	REF_NO2	CHAR	30	150:179	
	REF_NO3	CHAR	30	180:209	
	REF_NO4	CHAR	30	210:239	
	REASON_CODE	CHAR	6	240:245	
	VENDOR_NO	CHAR	10	246:255	
	VENDOR_INVC_NO	CHAR	30	256:285	
	PAYMENT_REF_NO	CHAR	16	286:301	
	PROOF_OF_DELIVERY_NO	CHAR	30	302:331	
	STATUS	CHAR	6	332:337	
	VALUE	CHAR	22	338:359	Includes an optional negative sign and a decimal point
	POS_TRAN_IND	CHAR	1	360:360	
	UPDATE_ID	CHAR	30	361:390	

Table Name	Column Name	Field Type	Field Width	Position	Description
	UPDATE_DATETIME	DATE	14	391:404	Format is YYYYMM DDHH24MI SS
	ERROR_IND	CHAR	1	405:405	
	BANNER_NO	INTEGER EXTERNAL	4	406:409	
	ROUND_AMT	INTEGER EXTERNAL	22	410:431	
	ROUNDED_OFF_AMT	INTEGER EXTERNAL	22	432:453	
	CREDIT_PROMOTION_ID	INTEGER EXTERNAL	10	454:463	
	REF_NO25	CHAR	30	464:493	
	REF_NO26	CHAR	30	494:523	
	REF_NO27	CHAR	30	524:553	
	STORE	INTEGER EXTERNAL	10	554:563	
	DAY	INTEGER EXTERNAL	3	564:566	
	RTLOG_ORIG_SYS	CHAR	3	567:569	
	TRAN_PROCESS_SYS	CHAR	3	570:572	

File Name: Satitem.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_ITEM	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ITEM_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	ITEM_STATUS	CHAR	6	25:30	
	ITEM_TYPE	CHAR	6	31:36	
	ITEM	CHAR	25	37:61	
	REF_ITEM	CHAR	25	62:86	
	NON_MERCH_ITEM	CHAR	25	87:111	
	VOUCHER_NO	CHAR	25	112:136	
	DEPT	INTEGER EXTERNAL	4	137:140	

Table Name	Column Name	Field Type	Field Width	Position	Description
	CLASS	INTEGER EXTERNAL	4	141:144	
	SUBCLASS	INTEGER EXTERNAL	4	145:148	
	QTY	DECIMAL EXTERNAL	14	149:162	Includes an optional negative sign and a decimal point
	UNIT_RETAIL	DECIMAL EXTERNAL	21	163:183	Includes a decimal point
	UNIT_RETAIL_VAT_INCL	CHAR	1	184:184	Indicates whether unit retail includes or excludes VAT
	SELLING UOM	CHAR	4	185:188	
	OVERRIDE_REASON	CHAR	6	189:194	
	ORIG_UNIT_RETAIL	DECIMAL EXTERNAL	21	195:215	Includes a decimal point
	STANDARD_ORIG_UNIT_RETAIL	DECIMAL EXTERNAL	21	216:236	
	TAX_IND	CHAR	1	237:237	
	ITEM_SWIPED_IND	CHAR	1	238:238	
	ERROR_IND	CHAR	1	239:239	
	DROP_SHIP_IND	CHAR	1	240:240	
	WASTE_TYPE	CHAR	6	241:246	
	WASTE_PCT	DECIMAL EXTERNAL	12	247:258	Includes a decimal point
	PUMP	CHAR	8	259:266	
	RETURN_REASON_CODE	CHAR	6	267:272	
	SALESPERSON	CHAR	10	273:282	
	EXPIRATION_DATE	DATE	8	283:290	Format is YYYYMMDD
	STANDARD_QTY	DECIMAL EXTERNAL	14	291:304	Includes an optional negative sign and a decimal point
	STANDARD_UNIT_RETAIL	DECIMAL EXTERNAL	21	305:325	Includes a decimal point
	STANDARD_UOM	CHAR	4	326:329	
	REF_NO5	CHAR	30	330:359	
	REF_NO6	CHAR	30	360:389	

Table Name	Column Name	Field Type	Field Width	Position	Description
	REF_NO7	CHAR	30	390:419	
	REF_NO8	CHAR	30	420:449	
	CATCHWEIGHT_IND	CHAR	1	450:450	
	SELLING_ITEM	CHAR	25	451:475	
	CUSTOMER_ORDER_LINE_NO	INTEGER EXTERNAL	6	476:481	
	MEDIA_ID	INTEGER EXTERNAL	10	482:491	
	UOM_QUANTITY	INTEGER EXTERNAL	12	492:503	
	TOTAL_IGTAX_AMT	DECIMAL EXTERNAL		504:524	
	UNIQUE_ID	CHAR	25	525:652	
	STORE	INTEGER EXTERNAL	10	653:662	
	DAY	INTEGER EXTERNAL	3	663:665	
	CUST_ORDER_NO	CHAR	48	666:713	
	CUST_ORDER_DATE	DATE	14	714:727	Format is YYYYMMDDHH24MISS
	FULFILL_ORDER_NO	CHAR	48	728:775	
	NO_INV_RET_IND	CHAR	1	776:776	
	SALES_TYPE	CHAR	1	777:777	
	RETURN_WH	INTEGER EXTERNAL	10	778:787	
	RETURN_DISPOSITION	CHAR	10	788:797	

File Name: Sattend.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_TENDER	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	TENDER_SEQ_NO	INTEGER EXTERNAL	4	21:24	
	TENDER_TYPE_GROUP	CHAR	6	25:30	
	TENDER_TYPE_ID	INTEGER EXTERNAL	6	31:36	

Table Name	Column Name	Field Type	Field Width	Position	Description
	TENDER_AMT	DECIMAL EXTERNAL	22	37:58	Includes an optional negative sign and a decimal point.
	CC_NO	INTEGER EXTERNAL	40	59:98	
	CC_EXP_DATE	DATE	8	99:106	Format is YYYYMMDD
	CC_AUTH_NO	CHAR	16	107:122	
	CC_AUTH_SRC	CHAR	6	123:128	
	CC_ENTRY_MODE	CHAR	6	129:134	
	CC_CARDHOLDER_VERF	CHAR	6	135:140	
	CC_TERM_ID	CHAR	5	141:145	
	CC_SPEC_COND	CHAR	6	146:151	
	VOUCHER_NO	CHAR	25	152:167	
	COUPON_NO	CHAR	16	168:183	
	COUPON_REF_NO	CHAR	16	184:199	
	CHECK_ACCT_NO	CHAR	30	209:238	
	CHECK_NO	INTEGER EXTERNAL	10	239:248	
	IDENTI_METHOD	CHAR	6	249:254	
	IDENTI_ID	CHAR	40	255:294	
	ORIG_CURRENCY	CHAR	3	295:297	
	ORIG_CURR_AMT	DECIMAL EXTERNAL	22	298:319	
	REF_NO9	CHAR	30	320:349	
	REF_NO10	CHAR	30	350:379	
	REF_NO11	CHAR	30	380:409	
	REF_NO12	CHAR	30	410:439	
	ERROR_IND	CHAR	1	440:440	
	STORE	INTEGER EXTERNAL	10	441:450	
	DAY	INTEGER EXTERNAL	3	451:453	

File Name: Satpymt.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_PAYMENT	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	PAYMENT_SEQ_NO	INTEGER EXTERNAL	20	21:24	
	PAYMENT_AMT	DECIMAL EXTERNAL	5	25:46	
	ERROR_IND	CHAR	10	47:47	
	STORE	INTEGER EXTERNAL	6	48:57	
	DAY	INTEGER EXTERNAL	3	58:60	

File Name: Samisstr.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_MISSING_TRAN	MISS_TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	STORE_DAY_SEQ_NO	INTEGER EXTERNAL	20	21:40	
	REGISTER	CHAR	5	41:45	
	TRAN_NO	INTEGER EXTERNAL	10	46:55	
	STATUS	CHAR	6	56:61	
	RTLOG_ORIG_SYS	CHAR	3	62:64	

File Name: Sattax.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_TRAN_TAX	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	TAX_CODE	CHAR	6	21:26	
	TAX_SEQ_NO	INTEGER EXTERNAL	4	27:30	

Table Name	Column Name	Field Type	Field Width	Position	Description
	TAX_AMT	DECIMAL EXTERNAL	22	31:52	Includes an optional negative sign and a decimal point
	ERROR_IND	CHAR	1	53:53	
	REF_NO17	CHAR	30	54:83	
	REF_NO18	CHAR	30	84:113	
	REF_NO19	CHAR	30	114:143	
	REF_NO20	CHAR	30	144:173	
	STORE	INTEGER EXTERNAL	10	174:183	
	DAY	INTEGER EXTERNAL	3	184:186	

File Name: Sacustatt.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_CUST_ATTRIB	TRAN_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	ATTRIB_SEQSO	CHAR	4	21:24	
	ATTRIB_TYPE	CHAR	6	25:30	
	ATTRIB_VALUE	CHAR	6	31:36	
	STORE	INTEGER EXTERNAL	10	37:46	
	DAY	INTEGER EXTERNAL	3	47:49	

File Name: Saerror.ctl

Table Name	Column Name	Field Type	Field Width	Position	Description
SA_ERROR	ERROR_SEQ_NO	INTEGER EXTERNAL	20	1:20	
	STORE_DAY_SEQ_NO	INTEGER EXTERNAL	20	21:40	

Table Name	Column Name	Field Type	Field Width	Position	Description
	BAL_GROUP_SEQ_NO	INTEGER EXTERNAL	20	41:60	
	TOTAL_SEQ_NO	INTEGER EXTERNAL	20	61:80	
	TRAN_SEQ_NO	INTEGER EXTERNAL	20	81:100	
	ERROR_CODE	CHAR	25	101:125	
	KEY_VALUE_1	INTEGER EXTERNAL	4	126:129	
	KEY_VALUE_2	INTEGER EXTERNAL	4	130:133	
	REC_TYPE	CHAR	6	134:139	
	STORE_OVERRIDE_IND	CHAR	1	140:140	
	HQ_OVERRIDE_IND	CHAR	1	141:141	
	UPDATE_ID	CHAR	30	142:171	
	UPDATE_DATE TIME	DATE	14	172:185	Format is YYYYMMDDHH24MISS
	ORIG_VALUE	CHAR	70	186:255	
	STORE	INTEGER EXTERNAL	10	256:265	
	DAY	INTEGER EXTERNAL	3	266:268	

ReSA Interface File Layout [rtlog]

The following example illustrates the file layout format of the Oracle Retail TLOG. The content of each Oracle Retail TLOG file is per store per day. The filename convention is RTLOG_STORE_DATETIME.DAT (for example, RTLOG_1234_01221989010000.DAT).

Fields, credit promotion ID, tax (vat) at item level and payment amount of customer orders are rounded off. Both VAT and tax are handled.

```
FHEAD             (Only 1 per file, required)
    THEAD         (Multiple expected, one per transaction, required for each
transaction)
    TCUST      (Only 1 per THEAD record allowed, optional for some transaction
types, see table below)
       CATT    (Attribute record specific to the TCUST record – Multiple allowed,
only valid if TCUST exists)
```

```
        TITEM       (Multiple allowed per transaction, optional for some
transaction types, see table below)
            IDISC   (Discount record specific to the TITEM record – Multiple
allowed per item, optional see table below)
            IGTAX   (Vat/Tax record specific to the TITEM record – Multiple allowed
per item, optional. Either TTAX or IGTAX should appear in a given RTLOG, but not
both, see table below)
        TTAX        (Vat/Tax record specific to the THEAD record – Multiple allowed
per transaction, optional. Either TTAX or IGTAX should appear in a given RTLOG,
but not both, see table below)
    TPYMT (Multiple allowed per transaction, will have the deposit amount for
pickup/delivery/layaway orders, optional see table below)
        TTEND       (Multiple allowed per transaction, optional for some
transaction types, see table below)
TTAIL           (1 per THEAD, required)
FTAIL               (1 per file, required)
```

The order of the records within this transaction layout is important. It aids processing by ensuring that the information is present when it is needed.

Record Name	Field Name	Field Type	Default Value	Description	Required ?	Justification/Padding
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type.	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0
	File Type Definition	Char(4)	RTLG	Identifies file as Oracle Retail TLOG.	Y	Left/Blank
	File Create Date	Char(14)	Create date	Date and time file was written by external system (YYYYMMDDHHMMSS).	Y	Left/None
	Business Date	Char(8)	Business Date to process	Business date of transactions (YYYYMMDD).	Y	Left/None
	Location Number	Char(10)	Specified by external system	Store or warehouse identifier.	Y	Left/None
	Reference Number	Char(30)	Specified by external system	This may contain the Polling ID associated with the consolidated TLOG file or used for other purpose.	N	Left/Blank
	RTLOG Originating System	Char(3)	POS	Identifies the system the RTLOG file originated from. Valid values are OMS and POS.	Y	Left/None
Transaction Header	File Type Record Descriptor	Char(5)	THEAD	Identifies file record type.	Y	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/Padding
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0
	Register	Char(5)		Till used at the store.	Y	Left/Blank
	Transaction Date	Char(14)	Transaction date	Date for the transactions that were processed at the POS (YYYYMMDDHHMMSS).	Y	Left/None
	Transaction Number	Number(10)		Transaction identifier. If sa_system_options, wkstation_tran_append_ind is Y, the first 3 digits indicate the workstation ID and last 7 digits indicate the transaction number.	Y	Right/0
	Cashier	Char(10)		Cashier identifier.	N	Left/Blank
	Salesperson	Char(10)		Salesperson identifier.	N	Left/Blank
	Transaction Type	Char(6)	Refer to TRAT code_type for a list of valid types.	Transaction type.	Y	Left/Blank
	Sub-transaction type	Char(6)	Refer to TRAS code_type for a list of valid types.	Sub-transaction type. For sale, it can be employee, drive-off, and so on.	N	Left/Blank
	Orig_tran_no	Number(10)		Populated only for post-void transactions. Transaction number for the original transaction that will be cancelled.	N	Right/0
	Orig_reg_no	Char(5)		Populated only for post-void transactions. Register number from the original transaction.	N	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/Padding
	Reason Code	Char(6)	Refer to REAC code_type for a list of valid codes. If the transaction type is PAIDOU and the sub transaction type is MV or EV, the valid codes come from the non_merch_code_head table.	Reason entered by the cashier for some transaction types. Required for Paid In and Paid out transaction types, but can also be used for voids, returns, and so on.	N	Left/Blank
	Vendor Number	Char(10)		Supplier ID for a merchandise vendor paid out transaction; partner ID for an expense vendor paid out transaction.	N	Left/Blank
	Vendor Invoice Number	Char(30)		Invoice number for a vendor paid out transaction.	N	Left/Blank
	Payment Reference Number	Char(16)		The reference number of the tender used for a vendor payout. This could be the money order number, check number, and so on.	N	Left/Blank
	Proof of Delivery Number	Char(30)		Proof of receipt number given by the vendor at the time of delivery. This field is populated for a vendor paid out transaction.	N	Left/Blank
	Reference Number 1	Char(30)		Number associated with a particular transaction, for example, whether for a Store Conditions transaction. The SA_REFERENCE table defines what this field can contain for each transaction type.	N	Left/Blank
	Reference Number 2	Char(30)		Second generic reference number.	N	Left/Blank
	Reference Number 3	Char(30)		Third generic reference number.	N	Left/Blank
	Reference Number 4	Char(30)		Fourth generic reference number.	N	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required ?	Justification/Padding
	Value Sign	Char(1)	Refer to SIGN code_type for a list of valid codes.	Sign of the value.	Y if Value is present.	Left/None
	Value	Number(20)		Value, with 4 implied decimal places. Populated by the retailer for TOTAL transaction, populated by ReSA for SALE and RETURN transactions.	Y if tran is a TOTAL.	Right/0 when value is present. Blank when no value is sent.
	Banner id	Number(4)		Banner ID of the location.	Y	Right/0 when value is present. Blank when no value is sent
	Rounded Amount Sign	Char(1)	Refer to SIGN code_type for a list of valid codes.	Sign of rounded amount.	Y	Left/None
	Rounded Amount	Number(20)		Total rounded amount, with 4 implied decimal places.	Y	Right/0 when RoundedAmount is present otherwise blank
	Rounded Off Sign	Char(1)	Refer to SIGN code_type for a list of valid codes.		Y	Left/None
	Rounded Off Amount	Number(20)		Rounded off amount, with 4 implied decimal places	Y	Right/0 when RoundedAmount is present otherwise blank
	Credit Promotion Id	Char(10)		Credit Promotional ID.	Y	Left/None
	Reference Number 25	Char(30)			N	Left/Blank
	Reference Number 26	Char(30)			N	Left/Blank
	Reference Number 27	Char(30)			N	Left/Blank
	Transaction Processing System	Char(3)	Valid values are OMS and POS.	Contains the ID of the system that processed the transaction.	N	Left/None
Transaction Customer	File Type Record Descriptor	Char(5)	TCUST	Identifies the file record type.	Y	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required ?	Justification/Padding
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0
	Customer ID	Char(16)	Customer identifier	The ID number of a customer.	Y	Left/Blank
	Customer Type ID	Char(6)	Refer to CIDT code_type for a list of valid types.	Customer ID type.	Y	Left/Blank
	Customer Name	Char(120)		Customer name.	N	Left/Blank
	Address 1	Char(240)		Customer address.	N	Left/Blank
	Address 2	Char(240)		Additional field for customer address.	N	Left/Blank
	City	Char(120)		City.	N	Left/Blank
	State	Char(12)	State identifier	State.	N	Left/Blank
	Zip Code	Char(30)	Zip identifier	Zip code.	N	Left/Blank
	Country	Char(3)		Country.	N	Left/Blank
	Home Phone	Char(20)		Telephone number at home.	N	Left/Blank
	Work Phone	Char(20)		Telephone number at work.	N	Left/Blank
	E-mail	Char(100)		E-mail address.	N	Left/Blank
	Birthdate	Char(8)		Date of birth. (YYYYMMDD)	N	Left/Blank
Customer Attribute	File Type Record Descriptor	Char(5)	CATT	Identifies file record type.	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0
	Attribute type	Char(6)	Refer to SACA code_type for a list of valid types.	Type of customer attribute	Y	Left/Blank
	Attribute value	Char(6)	Refer to members of SACA code_type for a list of valid values.	Value of customer attribute.	Y	Left/Blank
Transaction Item	File Type Record Descriptor	Char(5)	TITEM	Identifies file record type.	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/Padding
	Item Status	Char(6)	Refer to SASI code_type for a list of valid codes.	Status of the item within the transaction. Valid values are: V – Void item S - Sold item R - Returned item ORI – Order Initiate ORC – Order Cancel ORD – Order Complete LIN – Layaway Initiate LCA – Layaway Cancel LCO – Layaway Complete	Y	Left/Blank
	Item Type	Char(6)	Refer to SAIT code_type for a list of valid codes.	Identifies what type of item is transmitted.	Y	Left/Blank
	Item number type	Char(6)	Refer to UPCT code_type for a list of valid codes.	Identifies the type of item number if the item type is ITEM or REF.	N	Left/Blank
	Format ID	Char(1)	VPLU format ID	Used to interpret VPLU items.	N	Left/Blank
	Item	Char(25)	Item identifier	Identifies the merchandise item.	N	Left/Blank
	Reference Item	Char(25)	Item identifier	Identifies the sub-transaction level merchandise item.	N	Left/Blank
	Non-Merchandise Item	Char(25)	Item identifier	Identifies a non-merchandise item.	N	Left/Blank
	Voucher	Char(25)		Gift certificate number.	N	Right/0
	Department	Number(4)		Identifies the department to which this item belongs. This is filled in by saimptlog.	N	Right/Blank
	Class	Number(4)	Class of the item	Class of item sold or returned. Not required from a retailer; populated by ReSA. This is filled in by saimptlog.	N	Right/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/Padding
	Subclass	Number(4)	Subclass of the item	Subclass of the item sold or returned. Not required from a retailer; populated by ReSA. This is filled in by saimptlog.	N	Right/Blank
	Quantity Sign	Char(1)	Refer to SIGN code_type for a list of valid codes.	Sign of the quantity	Y	Left/None
	Quantity	Number(12)		Number of items purchased, with 4 decimal places.	Y	Right/0
	Selling Unit of Measure	Char(4)		Unit of measure of the item's quantity.	Y	Left/None
	Unit Retail	Number(20)		Unit retail, with 4 implied decimal places.	Y	Right/0
	Override Reason	Char(6)	Refer to ORRC code_type for a list of valid codes.	This column is populated when an item's price has been overridden at the POS to define why it was overridden.	Y if unit retail was manually entered	Left/Blank
	Original Unit Retail	Number(20)		Value, with 4 implied decimal places. This column is populated when the item's price was overridden at the POS and the item's original unit retail is known.	Y if unit retail was manually entered	Right/0
	Taxable Indicator	Char(1)	Refer to YSNO code_type for a list of valid codes.	Indicates whether or not item is taxable.	Y	Left/None
	Pump	Char(8)		Fuel pump identifier.	N	Left/Blank
	Reference Number 5	Char(30)		Number associated with a particular item within a transaction, for example, special order number. The sa_reference table defines what this field can contain for each transaction type.	N	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required ?	Justification/Padding
	Reference Number 6	Char(30)		Second generic reference number at the item level.	N	Left/Blank
	Reference Number 7	Char(30)		Third generic reference number at the item level.	N	Left/Blank
	Reference Number 8	Char(30)		Fourth generic reference number at the item level.	N	Left/Blank
	Item_swiped_ind	Char(1)	Refer to YSNO code_type for a list of valid codes.	Indicates if the item was automatically entered into the POS system or if it had to be manually keyed.	Y	Left/None
	Return Reason Code	Char(6)	Refer to SARR code_type for a list of valid codes.	The reason an item was returned.	N	Left/Blank
	Salesperson	Char(10)		The salesperson who sold the item.	N	Left/Blank
	Expiration_date	Char(8)		Gift certificate expiration date (YYYYMMDD).	N	
	Drop Ship Ind	Char(1)	Refer to YSNO code type for a list of valid codes.	Indicates whether the item is part of a drop shipment.	Y	Left/None
	Uom_qty	Number(12)		Quantity of items purchased in the given UOM, with 4 decimal places.	Y	Right/0
	Catchweight_ind	Char(1)	Valid values are Y and N.	Identifies if the item is a catchweight item.		Left/None
	Selling item	Char(25)	Item identifier	Identifies the selling item.	N	Left/Blank
	Customer order line no	Number(6)		Identifies the customer order number.	N	Left/Blank
	Media id	Number(10)		Identifies the customer media ID.	N	Left/Blank
	Total Igtax Amount	Number(21)		Contains the Igtax amount.	N	Right/0
	Unique ID	Char(128)			N	Left/Blank
	Customer Order Number	Char(48)		Contains the customer order ID.	N	Left/None

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/Padding
	Customer Order Date	Char(14)		Contains the customer order date. Format is: YYYYMMDDHHMMSS	N	Left/Blank
	Fulfillment Order Number	Char(48)		Contains the order ID of the fulfillment order.	N	Left/None
	No Inventory Return	Char(1)		Indicates if there is an associated inventory with the return transaction with an External Customer Order sales type.	N	Left/Blank
	Sales Type	Char(1)		Indicates if the transaction is an In Store Customer Order, External Customer Order, or Regular Sale.	Y	Left/Blank
	Return Warehouse	Char(10)		Contains the ID of the physical warehouse to which the inventory is returned.	N	Left/Blank
	Return Disposition	Char(10)		Contains the return disposition of the returned items.	N	Left/Blank
Item Discount	File Type Record Descriptor	Char(5)	IDISC	Identifies the file record type.	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0
	RMS Promotion Number	Char(6)	Refer to PRMT code_type for a list of valid types.	The RMS promotion type.	Y	Left/Blank
	Discount Reference Number	Number(10)		Discount reference number associated with the discount type. For example, if the discount type is a promotion, this contains the promotion number.	N	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required ?	Justification/Padding
	Discount Type	Char(6)	Refer to SADT code_type for a list of valid types.	The type of discount within a promotion. This allows a retailer to further break down coupon discounts within the In-store promotion, for example.	N	Left/Blank
	Coupon Number	Char(16)		Number of a store coupon used as a discount.	Y if coupon	Left/Blank
	Coupon Reference Number	Char(16)		Additional information about the coupon, usually contained in a second bar code on the coupon.	Y if coupon	Left/Blank
	Quantity Sign	Char(1)	Refer to SIGN code_type for a list of valid codes.	Sign of the quantity.	Y	Left/None
	Quantity	Number(12)		The quantity purchased for which the discount is applied, with 4 implied decimal places.	Y	Right/0
	Unit Discount Amount	Number(20)		Unit discount amount for this item, with 4 implied decimal places.	Y	Right/0
	Reference Number 13	Char(30)		Number associated with a particular transaction type at the discount level. The sa_reference table defines what this field can contain for each transaction type.	N	Left/Blank
	Reference Number 14	Char(30)		Second generic reference number at the discount level.	N	Left/Blank
	Reference Number 15	Char(30)		Third generic reference number at the discount level.	N	Left/Blank
	Reference Number 16	Char(30)		Fourth generic reference number at the discount level.	N	Left/Blank
	Uom_qty	Number(12)		Quantity of items purchased in the given UOM with 4 decimal places.	Y	Right/0

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/Padding
	Catchweight_ind	Char(1)	Valid values are Y and N.	Identifies if the item is a catchweight item.		Left/None
	Promo component	Number(10)		If the discount is a promotion, this field contains the promotion component value associated with the promotion (discount reference number).	N	Left/Blank
Item Tax	File Type Record Descriptor	Char(5)	IGTAX	Identifies the file record typ.e	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0
	Tax Authority	Number(10)			Y	Left/0
	Igtax Code	Char(6)	Refer to tax_code/vat_code of tax_codes/vat_codes tables.	IGtax code (tax code/VAT code) to represent whether it is a State, City, or some other tax code/VAT code.	Y	Left/Blank
	Igtax Rate	Number(20)		Igtax rate, with 4 implied decimal places.	Y	Right/0
	Igtax Amount Sign	Char(1)	Refer to SIGN code_type for a list of valid codes.	Sign of the Igtax amount.	Y	Left/None
	Igtax Amount	Number(21)		Total igtax amount for this item, with 5 implied decimal places.	Y	Right/0
	Reference Number 21	Char(30)			N	Left/None
	Reference Number 22	Char(30)			N	Left/None
	Reference Number 23	Char(30)			N	Left/None
	Reference Number 24	Char(30)			N	Left/None
Transaction Tax	File Type Record Descriptor	Char(5)	TTAX	Identifies the file record type.	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/Padding
	Tax Code	Char(6)	Refer to TAXC code_type for as list of valid types.	Tax code (tax code/VAT code) to represent whether it is a State, City, or some other tax code/VAT code.	Y	Left/Blank
	Tax Sign	Char(1)	Refer to SIGN code_type for a list of valid codes.	Sign of the tax amount.	Y	Left/None
	Tax Amount	Number(20)		Total rax amount for this item, with 4 implied decimal places.	Y	Right/0
	Reference Number 17	Char(30)			N	Left/None
	Reference Number 18	Char(30)			N	Left/None
	Reference Number 19	Char(30)			N	Left/None
	Reference Number 20	Char(30)			N	Left/None
Transaction payment	File Type Record Descriptor	Char(5)	TPYMT	Identifies the file record type.	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0
	Payment Sign	Char(1)	Refer to SIGN code_type for a list of valid codes.	Sign of the deposit amount.	Y	Left/None
	Payment Amount	Number(20)		Deposit amount paid, with 4 implied decimal places.	Y	Right/0
Transaction Tender	File Type Record Descriptor	Char(5)	TTEND	Identifies the file record type.	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0
	Tender Type Group	Char(6)	Refer to TENT code_type for as list of valid types	High-level grouping of tender types.	Y	Left/Blank
	Tender Type ID	Number(6)	Refer to the pos_tender_type_head table for as list of valid types.	Low-level grouping of tender types.	Y	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required ?	Justification/Padding
	Tender Sign	Char(1)	Refer to SIGN code_type for a list of valid codes.	Sign of the value.	Y	Left/None
	Tender Amount	Number(20)		Amount paid with this tender in the transaction, with 4 implied decimal places.	Y	Right/0
	Cc_no	Char(40)		Credit card number.	Y if credit card	Left/Blank
	Cc_auth_no	Char(16)		Authorization number for a credit card.	Y if credit card	Left/Blank
	cc authorization source	Char(6)	Refer to CCAS code_type for as list of valid types.		Y if credit card	Left/Blank
	cc cardholder verification	Char(6)	Refer to CCVF code_type for as list of valid types.		Y if credit card	Left/Blank
	cc expiration date	Char(8)		YYYYMMDD	Y if credit card	Left/Blank
	cc entry mode	Char(6)	Refer to CCEM code_type for as list of valid types.	Indicates whether the credit card was swiped, thus automatically entered, or manually keyed.	Y if credit card	Left/Blank
	cc terminal id	Char(5)		Terminal number from which the transaction was sent.	N	Left/Blank
	cc special condition	Char(6)	Refer to CCSC code_type for as list of valid types.		Y if credit card	Left/Blank
	Voucher_no	Char(25)		Gift certificate or credit voucher serial number.	Y if voucher	Right/0
	Coupon Number	Char(16)		Number of a manufacturer's coupon used as a tender.	Y if coupon	Left/Blank
	Coupon Reference Number	Char(16)		Additional information about the coupon, usually contained in a second bar code on the coupon.	Y if coupon	Left/Blank
	Cheque Account Number	Char(30)		Account number of the check.	N	Left/Blank

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/Padding
	Cheque Number	Number(10)		Check number.	Required for the tender type CHECK	Right/0
	Identification Method	Char(6)	Refer to IDMH code_type for list of valid types.	Identification Method (such as a driver's license number or photo credit card).	N	Left/Blank
	Identification Id	Char(40)		Identification ID (license ID or photo card number).	N	Left/Blank
	Original Currency	Char(3)	Refer to the CURRENCIES table for valid currency codes.	The original currency with which the customer made the payment.	N	Left/Blank
	Original Currency Amount	Number(20)		Amount paid with this tender in the original currency, with 4 implied decimal places.	N	Right/0
	Reference No 9	Char(30)		Number associated with a particular transaction type at the tender level. The sa_reference table defines what this field can contain for each transaction type.	N	Left/Blank
	Reference No 10	Char(30)		Second generic reference number at the tender level.	N	Left/Blank
	Reference No 11	Char(30)		Third generic reference number at the tender level.	N	Left/Blank
	Reference No 12	Char(30)		Fourth generic reference number at the tender level.	N	Left/Blank
Transaction Trailer	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type.	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0

Record Name	Field Name	Field Type	Default Value	Description	Required?	Justification/Padding
	Transaction Record Counter	Number(10)		Number of records processed in the current transactio (only those records between transaction head and tail).		
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies the file record type.	Y	Left/Blank
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Y	Right/0
	File Record Counter	Number(10)		Number of transactions processed in the current file (only the records between the file head and tail).	Y	Right/0

The RTLOG file is imported into the Sales Audit tables after validation by the batch program saimptlog. This section describes the requirements and validations performed on the records.

Common Requirements/Validations

This section details the common requirements and validations performed on all transactions. The following sections describe the specific requirements of each type of transaction. If a transaction is not mentioned, it does not have specific requirements.

Record Type Requirements:

Transaction Type	Includes item records?	Includes tender records?	Includes tax records? IGTAX?	Includes customer records?
OPEN	No	No	No	No
NOSALE	No	Optional	No	No
VOID	Optional	Optional	Optional	Optional
PVOID	No	No	No	No
SALE	Yes	Yes	Optional	Optional
RETURN	Yes	Yes	Optional	Optional
EEXCH	Yes	No	Optional	Optional
PAIDIN	No	*Yes*	No	No
PAIDOU	No	Yes	No	No
PULL	No	Yes	No	No
LOAN	No	Yes	No	No
COND	No	No	No	No
CLOSE	No	*No*	No	No

Transaction Type	Includes item records?	Includes tender records?	Includes tax records? IGTAX?	Includes customer records?
TOTAL	No	No	No	No
REFUND	This transaction is not sent through the RTLOG. It is entered at the HQ level. The TITEM and TCUST records are optional. The TTEND record is required. A TTAX record should not be included if IGTAX appears in a transaction. IGTAX is an item-level tax and TTAX is a transaction-level tax. Either IGTAX or TTAX can be used, but not both.			
METER	*Yes*	No	No	No
PUMPT	Yes	No	No	No
TANKDP	Yes	No	No	No
TERM	TERM records are created by saimptlog and then loaded into the database. They do not come from the RTLOG file. They require one TITEM, one TTEND, one TTAX, one TCUST record, one CATT record, IGTAX, and one TPYMT.			
DCLOSE	No	No	No	No
SPLORD	Optional	Yes	Optional	Optional

Requirements per record type:

Record Type	Requirements
IDISC	IDISC records must immediately follow their associated TITEM record.
IGTAX	IGTAX will immediately follow TITEM if IDISC is not coming, otherwise it should follow IDISC. Even if IGTAX is coming prior to IDISC, it will be processed, but for maintaining proper format, ReSA expects it to come after IDISC.
TTAX	Either this record or IGTAX should appear in the transaction. IGTAX and TTAX cannot be both used at the same time.
TPYMT	This record should be right before the TTEND record. It contains the deposit amount for pickup/delivery/layaway orders.
CATT	CATT records must immediately follow their associated TCUST record.

Code Type Validations:

Record Name	Field Name	Code Type
Transaction Header	Transaction Type	TRAT
	Sub-transaction Type	TRAS
	Reason Code	REAC or values from non_merch_code_head if the transaction type is PAIDOU and the sub-transaction type is MV or EV.
	Value Sign	SIGN
	Vender No	If the transaction type is PAIDOU and the sub-transaction type is MV, this field is validated against the supplier table. If the transaction type is PAIDOU and the sub-transaction type is EV, this field is validated against the partner table.

Record Name	Field Name	Code Type
	Transaction Processing System	TSYS
Transaction Item	Item Type	SAIT
	Item Status	SASI
	Item Number Type	UPCT
	Quantity Sign	SIGN
	Taxable Indicator	YSNO
	Price Override Reason Code	ORRC
	Item Swiped Indicator	YSNO
	Sales Type	SASY
	Return Disposition	INV_STATUS_CODES table
	No Inventory Return	YSNO
	Return Reason Code	SARR
Item Discount	RMS Promotion Type	PRMT
	Discount Type	SADT
	Quantity Sign	SIGN
Transaction Customer	Customer ID Type	CIDT
Customer Attribute	Attribute Type	SACA
	Attribute value	Code types from the codes in SACA.
Transaction Tax	Tax code	TAXC from the CODE_DETAIL table or VATC from the VAT_CODES table.
	Tax sign	SIGN
Transaction Payment	Payment (Deposit Amount) Sign	SIGN
Transaction Tender	Tender Type Group	TENT
	Tender Sign	SIGN
	Tender Type ID	Pos_tender_type_head table
	CC Authorization Source	CCAS
	CC Cardholder Verification	CCVF
	CC Entry Mode	CCEM
	CC Special Condition	CCSC

The following dates are validated: Business Date, Transaction Date, and Expiration Date. Also, saimptlog accepts only business dates that are within the PERIOD.VDATE minus the SA_SYSTEM_OPTIONS.DAYS_POST_SALE value.

The store number is validated against the STORE table. Numeric fields are checked for non-numeric characters.

For transactions of type SALE, RETURN, and EEXCH, saimptlog checks whether a transaction is in balance:

- When TAX is on in the system (system_options.default_tax_type equals SALES):

  Transaction Items (Unit Retail * Unit Retail Sign * Quantity) of items which are on Regular Sale, Return, or EEXCH

  + Item Discounts (Unit Discount Amount * Unit Discount Sign * Quantity) of items which are on Regular Sale, Return, or EEXCH

  + Item Level Tax (Total Igtax Amount) of items which are on Regular Sale, Return, or EEXCH

  + Transaction Tax (Tax Amount * Tax Sign)

  + Transaction payment (Payment Amount * Payment Sign)

  equals Transaction Tenders (Tender Amount * Tender Sign)

  saimptlog will populate the Value field (on THEAD) with the transaction's sales value (item value minus discount value plus tax value) from the preceding calculation if it was not provided in the RTLOG. The following change is made in the sale total balancing: Value field in THEAD will be: (item value – discount value + tax value) for items which are on Regular Sale, Return, or EEXCH + payment value.

  Note: If this Value field is being used in creating some totals, then accordingly, these totals needs to be modified to accommodate the extra amount coming in.

- When VAT is on in the system (system_options.default_tax_type in GTAX, SVAT), look for other system options, along with class level and store level VAT indicators, which tell whether the unit retail is inclusive or exclusive of VAT. The logic of balancing will vary:

  Transaction Items (Unit Retail * Unit Retail Sign * Quantity) of items which are on Regular Sale, Return, or EEXCH

  + Item Discounts (Unit Discount Amount * Unit Discount Sign * Quantity) of items which are on Regular Sale, Return, or EEXCH

  + Item Level Tax (Total Igtax Amount) of items which are on Regular Sale, Return, or EEXCH (when VAT is off at the item level).

  + Transaction Tax (Tax Amount * Tax Sign)

  + Transaction payment (Payment Amount * Payment Sign)

  equals Transaction Tenders (Tender Amount * Tender Sign)

Vouchers are treated as follows:

- If an item sold is as a gift certificate (Transaction Item, Voucher field has a value), the issued information is written to the SA_VOUCHER table.
- If the Transaction Type is RETURN, and the Transaction Tender Type Group is voucher (VOUCH), the issued information is written to the SA_VOUCHER table.

- If the Transaction Type is SALE and the Transaction Tender Type Group is a voucher (VOUCH), the redeemed information is written to the SA_VOUCHER table.
- When a gift certificate is sold, the customer information should always be included. A receiving customer name value should be populated in the ref_no5 field, receiving customer state value should be populated in the ref_no6 field, and receiving customer country should be populated in the ref_no7 field. These reference fields can be changed by updating the sa_reference table, but the code needs to be modified as well. The expiration date is put in the expiration_date field in the TITEM record.

Other validations and points to consider:

- The salesperson in the TITEM record takes precedence over the salesperson in the THEAD record.
- If an item sold is a sub-transaction (REF) item (Transaction Item, reference item field has a value and item does not), it will be converted to the corresponding transaction level item (ITEM).
- If an item sold is an ITEM (Transaction Item, item field has a value), it will be validated against the RMS item tables.
- The corresponding Department, Class, Subclass, and Taxable Indicator will be selected from the RMS tables and populated for an item.

The balancing level determines whether the register or the cashier fields are required:

- If the balancing level is R (register), the register field on the THEAD must be populated.
- If the balancing level is C (cashier), the cashier field on the THEAD must be populated.
- If the balancing level is S (store), neither field is required to be populated.
- The tax_ind and the item_swiped_ind fields can only accept Y or N values. If an invalid value is passed through the RTLOG, an error will be flagged and the value will be defaulted to Y.

Transaction of Type SALE

A transaction of type SALE is generated whenever an item is sold. If a sale is for an employee, the sub-transaction type is EMP. If is is a drive-off sale, when someone drives off with unpaid gas, the sub-transaction type is DRIVEO. A special type of sale is an *odd exchange*, sub-transaction type EXCH, where items are sold and returned in the same transaction. If the net value of the exchange is positive, then it is a sale. If the net value is negative, it is a return.

Requirements per record type (other than what is described in the preceding Layout section):

Record Type	Requirements
THEAD	
TITEM	▪ Item Status is a required field; it determines whether the item is Sold (S), Returned (R), or Voided (V). If the item status is S, the quantity sign is expected to be P. If the item status is R, the quantity sign is expected to be N. Also, if the item status is ORI, LIN, ORD, or LCO, the quantity sign should be P. In the case of ORC or LCA, it should be N. ▪ If the item status is V, the quantity sign is the reverse of the quantity sign of the voided item. That is, if an item with status S is voided, the quantity sign would be N. Furthermore, the sum of the quantities being voided cannot exceed the sum of the quantities that are Sold or Returned. **Note:** Neither of the two validations are performed by saimptlog, but an audit rule could be created to check this. ▪ The following item statuses are used for handling items on customer order layaway: ORI – Order Initiate ORD – Order Complete ORC – Order Cancel LIN – Layaway Initiate LCA – Layaway Cancel LCO – Layaway Complete ▪ In a typical sale, the items all have a status of S. In the case of an odd exchange, some items will have a status of R. ▪ In a typical return, the items all have a status of R. In the case of an odd exchange, some items will have a status of S. ▪ If an item has status R, then the Return Reason Code field may be populated. If it is, it will be validated against code type SARR. Also, it is better to capture the Return Reason Code in the case of items on ORC or LCA, but it is not mandatory. No validation is kept for these new item statuses for checking of SARR. ▪ If the price of an item is overridden, the Override Reason and Original Unit Retail fields must be populated.
IDISC	▪ The RMS Promotion Type field must always be populated with values of code type PRMT. ▪ The Promotion field is validated, when a value is passed, against the promhead table. ▪ If the promotion is In Store (code 1004), the Discount Type field must be populated with values of code type SADT. ▪ The Discount Reference Number is a promotion number which is of status A, E, or M. ▪ If the Discount Type is SCOUP for Store Coupon, the Coupon Number field must be populated. The Coupon Reference Number field is optional.
IGTAX	▪ The IGTAX_CODE field must always be populated depending on the system's default tax type. For a default tax type of SALES, this field will be populated with values of code_type TAXC. For a default tax type of SVAT or GTAX,, this field will be populated with VATC (vat_code from vat_codes table). IGTAX is an Item-level tax. ▪ The TAX_AUTHORITY field must always be populated when the default tax type is GTAX.
TTAX	▪ The TAX_CODE field must always be populated depending on the system's default tax type. For a default tax type of SALES, this field will be populated with values of code_type TAXC. For a default tax type of GTAX or SVAT, this field will be populated with VATC (vat_code from vat_codes table). TTAX is a Transaction-level tax.

Record Type	Requirements
TPYMT	Payment (Deposit amount) sign and Payment (Deposit) amount fields are necessary if this line is appearing. Basically, this is the accumulation of various items being considered in one transaction, which are on pick up/delivery/lay away.
TTEND	If the tender type group is COUPON, the Coupon Number field must be populated. The Coupon Reference Number field is optional.

Meaning of reference number fields:

Note: The meaning of these reference number fields may be changed through the sa_reference table. The transaction type SPLORD is the same as SALE, but the inventory will not be reserved for the orders at its line level.

Transaction Type	Sub-transaction Type	Item Type	Tender Type Group	Reference Number Field	Meaning of Reference Field	Req?
SALE				1	Speed Sale Number	Y
SALE		GCN		5	Recipient Name	N
SALE		GCN		6	Recipient State	N
SALE		GCN		7	Recipient Country	N
SALE			CHECK	9	Check Number	N
SALE			CHECK	10	Driver's License Number	N
SALE			CHECK	11	Credit Card Number	N
SALE	DRIVEO			1	Incident Number	Y
SALE	EMP			3	Employee Number of the employee receiving the goods.	N

Expected values for sign fields:

TRANSACTION TYPE	TITEM.Quantity Sign	TTEND.Tender Sign	TTAX.Tax Sign	IDISC.Quantity Sign
SALE	P if item is sold; N if item is returned; reverse of original item if item is voided.	P	P	P if item is sold; N if item is returned; reverse of original item if item is voided.
SALE	P if item is on ORI, LIN, ORD, or LCO; N if item is on ORC or LCA.	P	P	P if item is on ORI, LIN, ORD, or LCO; N if item is on ORC or LCA.

Transaction of Type PVOID

This transaction is generated at the register when another transaction is being post voided. The orig_tran_no and orig_reg_no fields must be populated with the appropriate information for the transaction being post voided. The PVOID transaction must be associated with the same store day as the original transaction. If the PVOID needs to be generated after the store day is closed, the transaction needs to be created using the forms.

Transaction of type RETURN

This transaction is generated when a customer returns an item.

This type of transaction has similar record type requirements as a SALE transaction.

Meaning of reference number fields:

> **Note:** The assumption is that new item statuses will not come under transaction type RETURN.

If a customer wants to return the items (ORI, LIN), these will come under SALE but with item statuses as ORC or LCA.

> **Note:** The meaning of these reference number fields may be changed through the sa_reference table.

Transaction Type	Sub-transaction Type	Reference Number Field	Meaning of Reference Field	Req?
RETURN		1	Receipt Indicator (Y/N)	Y
RETURN		2	Refund Reference Number	N
RETURN	EMP	3	Employee Number of the employee returning the goods.	N

Expected values for sign fields:

TRANSACTION TYPE	TITEM.Quantity Sign	TTEND.Tender Sign	TTAX.Tax Sign	IDISC.Quantity Sign
RETURN	P if item is sold; N if item is returned; reverse of original item if item is voided.	N	N	P if item is sold; N if item is returned; reverse of original item if item is voided.

Transaction of type SPLORD

This transaction is generated when a customer picks up an item, which is not in stock. The item status can be ORI, ORC, or ORD. (Order Initiate, Order Cancel, or Order Complete).

Transaction of type EEXCH

This transaction is generated when there is an even exchange.

This type of transaction has similar record type requirements as a SALE transaction.

It is expected that the number of items returned equals the number of items sold. However, this validation is not performed by saimptlog. An audit rule could be created for this. Saimptlog only expects that there would be at least two item records.

No tender changes hands in this transaction.

Meaning of reference number fields:

Note:

The items, which are on customer order or layaway, should not be come under this transaction type.

The meaning of these reference number fields may be changed through the sa_reference table.

Transaction Type	Sub-transaction Type	Reference Number Field	Meaning of Reference Field	Req?
EEXCH		1	Receipt Indicator (Y/N)	Y
EEXCH	EMP	3	Employee Number of the employee exchanging the goods.	N

Transaction of Type PAIDIN

This type of transaction has only one TTEND record.

A reason code is required.

Meaning of reference number fields:

Note: The meaning of these reference number fields may be changed through the sa_reference table.

Reason Code	Reference Number Column	Meaning	Req?
NSF	1	NFS Check Credit Number	N
ACCT	1	Account Number	N

Transaction of Type PAIDOU

This type of transaction has only one TTEND record.

A reason code is required (code type REAC). If the sub-transaction type is EV or MV, the reason code comes from the non_merch_codes_head table.

If the sub-transaction type is EV or MV, then at least one field among the vendor number, vendor invoice number, payment reference number, and proof of delivery number fields should be populated.

If the sub-transaction type is EV, the vendor number comes from the partner table. If the sub-transaction type is MV, the vendor number comes from the supplier table.

Meaning of reference number fields:

> **Note:** The meaning of these reference number fields may be changed through the sa_reference table.

Sub Transaction Type	Reason Code	Reference Number Column	Meaning	Req?
EV		2	Personal ID Number	N
EV		3	Routing Number	N
EV		4	Account Number	N
	PAYRL	1	Money Order Number	N
	PAYRL	2	Employee Number	N
	INC	1	Incident Number	N

Transaction of Type PULL

This transaction is generated when cash is withdrawn from the register.

This type of transaction has only one TTEND record.

Expected values for sign fields:

TRANSACTION TYPE	TITEM.Quantity Sign	TTEND.Tender Sign	TTAX.Tax Sign	IDISC.Quantity Sign
PULL	N/A	N	N/A	N/A

Transaction of Type LOAN

This transaction is generated when cash is added to the register.

This type of transaction has only one TTEND record.

Expected values for sign fields:

TRANSACTION TYPE	TITEM.Quantity Sign	TTEND.Tender Sign	TTAX.Tax Sign	IDISC.Quantity Sign
LOAN	N/A	P	N/A	N/A

Transaction of Type COND

This transaction records the condition at the store when it opens. There can be at most one COND record containing weather information and at most one COND record containing temperature information. Both of these pieces of information may be in the same COND record. There may be any number of COND records containing traffic and construction information.

This type of transaction does not have TITEM, IDISC, IGTAX, TTAX, TPYMT, or TTEND records.

Note: The meaning of these reference number fields may be changed through the sa_reference table.

Reference Number Column	Meaning	Req?
1	Weather – code type WEAT	N
2	Temperature – a signed 3 digit number.	N
3	Traffic – code_type TRAF	N
4	Construction – code_type CONS	N

Transaction of Type TOTAL

This transaction records the totals that are reported by the POS and OMS. The value field must be populated. Some systems generate only one transaction number for all totals. In order to avoid duplicate errors being reported, only one total transaction can have a transaction number and the subsequent ones can have blank transaction numbers. In other words, a TOTAL transaction is not required to have a transaction number.

This type of transaction does not have TITEM, IDISC, IGTAX, TTAX, TPYMT, TTEND records.

Transaction of Type METER

This transaction is generated when a meter reading of a fuel pump is taken.

This type of transaction has only TITEM records.

Meaning of reference number fields:

Note: The meaning of these reference number fields may be changed through the sa_reference table.

Reference Number Column	Meaning	Req?
1	Reading Type: (A for adjustment, S for shift change, P for price change, or C for store close)	Y
5	Opening Meter Readings	Y
6	Closing Meter Reading	Y
7	If the reading type is P for price change, the old unit retail should be placed here. Decimal places are required.	Y
8	Closing Meter Value	Y

Transaction of Type PUMPT

This transaction is generated when a pump test is performed. This type of transaction has only TITEM records.

Transactions of Type TANKDP

This transaction is generated when a tank dip measurement is taken.

This type of transaction has only TITEM records.

Meaning of reference number fields:

> **Note:** The meaning of these reference number fields may be changed through the sa_reference table.

Reference Number Column	Meaning	Req?
1	Tank identifier	Y
5	Dip Type (FUEL, WATER, and so on)	Y
6	Dip Height Major (decimal places required)	Y
7	Dip Height Minor (decimal places required)	Y

Transaction of Type DCLOSE

This transaction is generated when the day closed. The transaction number for this type of transaction has to be blank.

> **Note:** Vouchers are minimally handled by saimptlog. Voucher information is written to the savouch file which is passed to the program savouch.pc.

- A voucher will appear on the TITEM record only if it was sold. When saimptlog encounters a SALE transaction with a voucher, it writes the voucher to the savouch file as an I for Issued voucher.
- A voucher will be issued when it appears on the TTEND record of transactions of type RETURN and PAIDOU. In other words, saimptlog will write it to the savouch file with status I.
- A voucher will be redeemed when it appears on the TTEND record of transactions of type SALE and PAIDIN. In other words, saimptlog will write it to the savouch file with status R.

Vouchers may not be returned. However, a transaction of type PAIDOU may be generated when the customer exchanges a voucher for another form of tender.

Design Assumptions

ReSA Valid Transaction Types

Transaction Code	Transaction Type
OPEN	Open
CLOSE	Close
COND	Daily Store Conditions

Transaction Code	Transaction Type
DCLOSE	Day close indicator
LOAN	Loan
METER	Meter Reading for Fuel
NOSALE	No Sale
PAIDIN	Paid In
PAIDOU	Paid Out
PULL	Pull
PUMPT	Pump Test for Fuel
PVOID	Post Void (A transaction that was rung later into the register to void something that occurred earlier at the same store/day. A post void updates the original transaction's sub-transaction type.)
REFUND	Return of customer's original check.
RETURN	Return
SALE	Sale
TANKDP	Tank Dip
TOTAL	POS generated totals
EEXCH	Even exchange
VOID	Void (aborted transaction)

The DCLOSE Transaction Type

When the retailer is sending only one file to the system, SAIMPTLOG.PC marks the store day record in the ReSA import log as partially or fully loaded in the database by looking for a transaction type of DCLOSE. However, if the retailer is sending more than one file (as in, for example, a trickle polling situation), the retailer can specify the number of files that the system should expect in combination with the DCLOSE transaction type. This ensures that the system receives all of the files, even if the DCLOSE transaction type is, for some reason, received before the final file.

For example, if 24 files are expected over a given amount of time, and the file with the DCLOSE transaction type is, for some reason, sent before the 24th file, the RMS system waits until the last file arrives before marking the store day record as partially or fully loaded in the database.

The import process is completed after SAIMPTLOGFIN.PC has updated the store, data, and audit status of each store day record.

saimptlogtdup_upd (Processing to Allow Re-Upload of Deleted Transactions)

Module Name	saimptlogtdup_upd.pc
Description	Processing to Allow Re-Upload of Deleted Transactions
Functional Area	Oracle Retail Sales Audit
Module Type	Admin
Module Technology	ProC
Integration Catalog ID	RSA19

Design Overview

The purpose of this batch module is to fetch all deleted transactions for a store day and modify the tdup<Store><rtlog originating system>.dat file to remove deleted transactions, from the tdup range, in order to facilitate the saimptlog/saimptlogi batch to upload deleted transactions again. The batch will process all the store day with data status in Partially Loaded and Ready For Import and a business date that lies between the vdate minus the sa_syatem_options. day_post_sale and the vdate. The batch will not process a store day, if the tdup<Store><rtlog originating system>.dat file does not exist. The batch is designed to work only if sa_system_options.check_dup_miss_tran is set to Y, otherwise, do nothing and come out with successful completion. Also, the batch will not terminate with an error, if the deleted transaction to be removed from tdup range does not exist in the tdup<Store><rtlog originating system>.dat file.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Adhoc
Scheduling Considerations	This program should be run before running saimptlog/saimptlogi if any Store-Day's have been deleted.
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_STORE_DAY	Yes	No	Yes	No
SA_TRAN_HEAD	Yes	No	No	No

Integration Contract

Integration Type	NA
File Name	NA
Integration Contract	NA

Design Assumptions

NA

saimptlogfin (Complete Transaction Import Processing)

Module Name	saimptlogfin.pc
Description	Complete Transaction Import Processing
Functional Area	Oracle Retail Sales Audit
Module Type	Admin
Module Technology	ProC
Integration Catalog ID	RSA18

Design Overview

The saimptlogfin program creates the balances (over or under) by store, register, or cashier and populates it in the SA_BALANCE_GROUP table. It also cancels post voided transactions and vouchers and validates missing transactions. It marks the store day record in the ReSA import log as partially or fully loaded. This will unlock the store day records after all store transactions are imported.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	Run towards the end of loading POS transaction data into ReSA. It should run after SAIMPTLOG or SAIMPTLOGI and SAVOUCH, but before SATOTALS.PC.
Pre-Processing	Saimptlog/saimptlogi, savouch
Post-Processing	satotals
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table	Select	Insert	Update	Delete
STORE_DAY_SEQ_NO	Yes	No	No	No
STORE	Yes	No	No	No
SA_STORE_DAY	Yes	No	Yes	No

Table	Select	Insert	Update	Delete
SA_CUST_ATTRIB	Yes	No	No	Yes
SA_CUSTOMER	Yes	No	No	Yes
SA_TRAN_DISC	Yes	No	No	Yes
SA_TRAN_ITEM	Yes	No	No	Yes
SA_TRAN_TAX	Yes	No	No	Yes
SA_TRAN_TENDER	Yes	No	No	Yes
SA_ERROR	Yes	No	Yes	Yes
SA_MISSING_TRAN	Yes	No	No	Yes
SA_TRAN_HEAD	Yes	No	Yes	Yes
SA_BALANCE_GROUP	Yes	Yes	No	No
SA_TRAN_HEAD CANCEL	Yes	No	No	No
SA_STORE_DATA	Yes	No	No	No
SA_IMPORT_LOG	No	No	Yes	No
SA_TRAN_HEAD_REV	No	Yes	No	No
SA_TRAN_ITEM_REV	No	Yes	No	No
SA_TRAN_TENDER_REV	No	Yes	No	No
SA_TRAN_TAX_REV	No	Yes	No	No
SA_TRAN_DISC_REV	No	Yes	No	No
SA_ERROR_REV	No	Yes	No	No
SA_EXPORTED_REV	No	Yes	No	No

Integration Contract

Integration Type	NA
File Name	NA
Integration Contract	NA

Design Assumptions

NA

savouch (Sales Audit Voucher Upload)

Module Name	savouch.pc
Description	Sales Audit Voucher Upload
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA08

Design Overview

Because gift certificates can enter the Sales Audit system as either items or tender, processing must be done to match up the sales and redemptions. This module is used to aggregate gift certificate and voucher records. It compares records in the input files to the database. If a record for the voucher does not exist on the database, the record is inserted. If the voucher already exists on the database, the record should be updated with the appropriate information. The voucher details are updated to SA_VOUCHER table.

Some retailers assign gift certificates to a given store, which means that before a gift certificate is sold at a store, it is assigned to a given store. When a retailer assigns a gift certificate to a given store, a record is written to the database. When the gift certificate is then sold by the store and redeemed by the consumer, this existing record must be updated to include the sale and redemption information. Some retailers choose not to assign gift certificates and instead simply sell gift certificates. In that case, the record will be inserted into the database when the gift certificate is sold and then updated when the gift certificate is redeemed.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This program needs to be scheduled after either saimpltog.pc and its sqlldr process, or saimpltogi.pc, but before saimpltogfin.pc.
Pre-Processing	saimptlog/saimptlogi
Post-Processing	Saimptlogfin.pc

Restart/Recovery

Restart/recovery logic for file-based processing is used. Records will be committed to the database when the commit_max_ctr defined in the RESTART_CONTROL table is reached.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_VOUCHER	Yes	Yes	Yes	No

Integration Contract

Integration Type	Upload to ReSA
File Name	The input file name is not fixed; the input file name is determined by a runtime parameter. Records rejected by the import process are written to a reject file. The reject file name is not fixed; the reject file name is determined by a runtime parameter.
Integration Contract	IntCon000160 (SAVO)

Input File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor/	Char(5)	FHEAD	File head marker.
	Line id	Number(10)	0000000001	Unique line ID.
	Translator id	Char(5)	SAVO	Identifies transaction type.
	File create date	Char(14)		Vdate in YYYYMMDDHH24MISS format.
	Business Date	Char(8)	Business Date	Vdate in YYYYMMDD format.
FDETL	Record descriptor/	Char(5)	FDETL	File head marker.
	Line id	Number(10)		Unique line ID.
	Voucher seq Number	Number(20)		Unique identifier for an entry to the SA_VOUCHER table.
	Voucher No	Char(16)		Serial Number of the voucher.
	Tender Type Id	Number(6)		Type of Voucher (Valid values for tender type are maintained in the pos_tender_type_head table with tender_type_group as VOUCH.
	Assigned Date	Char(8)		Date the voucher was assigned.
	Assigned store	Number(10)		Store to which the voucher is assigned.
	Issuing Date	Char(8)		Date this document was issued.
	Issuing store	Number(10)		Store this document was issued from.
	Issuing Register	Char(5)		Register this document was issued from.

Record Name	Field Name	Field Type	Default Value	Description
	Issuing Cashier	Char(10)		Cashier issuing the document.
	Issued transaction number	Number(20)		Transaction number at the time of issuance.
	Issued item seq number	Number(4)		Will hold the item sequence of the item when the voucher is sold as an item (gift voucher).
	Issued tender seq number	Number(4)		Will hold the tender sequence of the tender when the voucher is sold as a tender (Merchandise Credit).
	Issued Amount	Number(20)		Amount the voucher was issued for*10000 (4 implied decimal places).
	Issued Customer Name	Char(120)		Name of the customer, who was issued the voucher.
	Issued Customer Addr1	Char(240)		The address of the customer who was issued the voucher.
	Issued Customer Addr2	Char(240)		The second line address of the customer who was issued the voucher.
	Issued Customer City	Char(120)		City of the customer, the voucher is issued.
	Issued Customer State	Char(3)		State of the customer.
	Issued Customer Postal Code	Char(30)		Postal address of the customer .
	Issued Customer Country	Char(3)		Country of the customer where the voucher was issued.
	Recipient Name	Char(120)		Name of the intended recipient.
	Recipient State	Char(3)		The state of the intended recipient.
	Recipient Country	Char(3)		The country of the intended recipient.
	Redemption Date	Char(8)		Date the voucher was redeemed.
	Redemption Store	Number(10)		Store at which the voucher was redeemed.
	Redemption Register	Char(5)		Register at which the document was redeemed.
	Redemption cashier	Char(10)		Cashier redeeming the voucher.
	Redemption tran seq number	Number(20)		Transaction Number when the document was redeemed.

Record Name	Field Name	Field Type	Default Value	Description
	Redemption Tender seq number	Number(4)		This column will hold the tender sequence of the tender within the transaction when a voucher is redeemed as tender.
	Redemption Amount	Number(20)		Amount the voucher was redeemed for*10000 (4 implied decimal places).
	Expiry Date	Char(8)		Expiry Date.
	Status	Char(1)		Indicator showing the document's status - issued or redeemed. Valid values = I – Issued, R - Redeemed.
	Comments	Char(2000)		Comments.
FTAIL	Record type	Char(5)	FTAIL	Describes file record and marks the end of file.
	Line id	Number(10)		Unique file line ID.
	#lines	Number(10)		Total number of transaction lines in file (not including FHEAD and FTAIL).

Design Assumptions

NA

saimpadj (Import Total Value Adjustments From External Systems to ReSA)

Module Name	saimpadj.pc
Description	Import Total Value Adjustments From External Systems to ReSA
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA07

Design Overview

This module posts external system adjustments to the Sales Audit total value table.

The sales audit adjustments are passed to the module in an external file.

Records that fail necessary validations would be written to the reject file. The input and reject file names are passed as arguments.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This module should be executed after the ReSA transaction import process and before the ReSA totaling process.
Pre-Processing	Saimptlogfin.pc
Post-Processing	Satotoals.pc
Threading Scheme	NA

Restart/Recovery

Restart/recovery logic for file-based processing is used. The logical unit of work for this module is a parameterized number defined in the restart tables.

Record level locking is done on sa_store_day before updating.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_TOTAL	Yes	No	No	No

Table	Select	Insert	Update	Delete
SA_HQ_VALUE	No	Yes	No	No
SA_STORE_DAY	Yes	No	Yes	No
SA_EXPORT_LOG	Yes	Yes	No	No
SA_TOTAL_USAGE	Yes	No	No	No

Integration Contract

Integration Type	Upload to ReSA
File Name	Determined by runtime parameters.
Integration Contract	IntCon000047

Input File

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type (the beginning of the input file).
	File Line Identifier	Number(10)	Sequential number	ID of the current line being read from input file.
	File head descriptor	Char(4)	IMPA	Describes file line type.
	Current date	Char(14)		File date in YYYYMMDDHH24MISS format.
FDETL	File Type Record Descriptor	Char(5)	FDETL	Identifies the file record type to upload a new deal header.
	File Line Identifier	Number(10)	Sequential number	ID of the current line being read from input file.
	Data source	Char(6)		Name of the external system that produced the file.
	New value sign	Char(1)		Sign(+/-) for the new value.
	New Value	Number(20)		Value for the total entered by Headquarters user*10000 (4 implied decimal places).
	Total seq no	Number(20)		Identifies the unique result set for this total ID, total revision, or store/day. Balancing group and index values.
	Store	Number(10)		Store number for a store/day combination.

Record Name	Field Name	Field Type	Default Value	Description
	Business Date	Char(8)		Date for store/day combination.
	Total id	Char(10)		ID to uniquely identify the total.
	Ref no 1	Char(30)		The first reference value based by which the total is grouped.
	Ref no 2	Char(30)		The second reference value based by which the total is grouped.
	Ref no 3	Char(30)		The third reference value based by which the total is grouped.
FTAIL	File Type record descriptor	Char(5)	FTAIL	Identifies the file record type (the end of the input file).
	File Line Identifier	Number(10)	Sequential number	ID of the current line being read from input file.
	File Record Counter	Number(10)	Sequential number	Number of records/transactions in the current file (only records between head and tail).

Design Assumptions

NA

satotals (Calculate Totals based on Client Defined Rules)

Module Name	satotals.pc
Description	Calculate Totals based on Client Defined Rules
Functional Area	Sales Audit, Totals
Module Type	Business Processing
Module Technology	ProC
Integration Catalog ID	RSA16

Design Overview

This module produces totals from user-defined total calculation rules. Totaling is integral to the sales auditing process. Totaling provides the values against which auditors can compare receipts. These comparisons find data errors that could be the result of either honest mistakes or fraud. Finding these mistakes during the sales auditing process prevents these errors from being passed on to merchandising and data warehouse systems. Totaling also provides quick access to other numeric figures about the day's sales transactions.

Totaling in ReSA is dynamic. ReSA automatically totals transactions based on calculation definitions that the retailer's users create using the online Totals Calculation Definition Wizard. In addition, the retailer is able to define totals that come from the POS, but that ReSA does not calculate. Whenever users create new calculation definitions or edit existing ones, they become part of the automated totaling process the next time that this process runs.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This program will run after the batch program saimptlogfin.pc and before sarules.pc.
Pre-Processing	Saimptlogfin.pc
Post-Processing	Sarules.pc
Threading Scheme	This program runs against a single store at a time.

Restart/Recovery

The logical unit of work for this program is a SA_STORE_DAY record. Records are committed to the database when the commit_max_ctr defined for SATOTALS on the RESTART_CONTROL table is reached. This program achieves inherent restart/recovery

due to the fact that store/day records that are processed will be updated to an audit_status of T for Totaled and will not be fetched by the driving cursor when the program restarts.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_STORE_DAY	Yes	No	Yes	No
SA_TOTAL	No	Yes	No	No
SA_TOTAL_HEAD	Yes	No	No	No
SA_ERROR	No	Yes	No	Yes
SA_ERROR_WKSHT	No	Yes	No	Yes
SA_POS_VALUE	No	Yes	No	No
SA_POS_VALUE_WKSHT	No	Yes	No	No
SA_SYS_VALUE	No	Yes	No	No
SA_SYS_VALUE_WKSHT	No	Yes	No	No
SA_ERROR_REV	No	Yes	No	No
SA_EXPORTED_REV	No	Yes	No	No
SA_EXPORTED	No	No	No	Yes

Integration Contract

Integration Type	NA
File Name	NA
Integration Contract	NA

Design Assumptions

NA

sarules (Evaluate Transactions and Totals based on Client Defined Rules)

Module Name	sarules.pc
Description	Evaluate Transactions and Totals based on Client Defined Rules
Functional Area	Oracle Retail Sales Audit
Module Type	Business Processing
Module Technology	ProC
Integration Catalog ID	RSA17

Design Overview

Evaluating rules is integral to the sales auditing process. Rules make the comparisons between data from various sources. These comparisons find data errors that could be the result of either honest mistakes or fraud. Finding these mistakes during the sales auditing process prevents these errors from being passed on to merchandising and data warehouse systems.

Rules in ReSA are dynamic. Aside from basic data validations, rules are not predefined in the system. Retailers have the ability to define them through the online Rule Definition Wizard. Errors uncovered by these rules are available for review online during the interactive audit process. After users modify existing rules or create new ones, they become part of the rules the next time that sarules.pc runs.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This program will run after SATOTALS and before SAESCHEAT. It should also be run before SAPREEXP and any of the ReSA export modules that extract store/day transaction data to other applications (for example, SAEXPRMS, SAEXPIM, SAEXPRDW, SAEXPACH, SAEXPUAR, and SAEXPGL).
Pre-Processing	satotoals
Post-Processing	Saescheat, sapreexp
Threading Scheme	This program runs against a single store at a time.

Restart/Recovery

The logical unit of work for this program is a SA_STORE_DAY record. Records are committed to the database when the commit_max_ctr defined for SARULES on the RESTART_CONTROL table is reached. This program achieves inherent restart/recovery due to the fact that store/day records that are processed will be updated to an audit_status of A (audited), H (HQ errors pending), or S (store errors pending) and will not be fetched by the driving cursor when the program restarts.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_STORE_DAY	Yes	No	Yes	No
SA_RULE_HEAD	Yes	No	No	No
SA_RULE_LOC_TRAIT	Yes	No	No	No
SA_ERROR_WKSHT	No	Yes	No	Yes
SA_ERROR_TEMP	No	Yes	No	No
SA_ERROR	No	Yes	Yes	Yes
SA_TOTAL	No	No	Yes	No
SA_TRAN_HEAD	No	No	Yes	No
SA_TRAN_ITEM	No	No	Yes	No
SA_TRAN_DISC	No	No	Yes	No
SA_TRAN_TENDER	No	No	Yes	No
SA_TRAN_TAX	No	No	Yes	No

Integration Contract

Integration Type	NA
File Name	NA
Integration Contract	NA

Design Assumptions

NA

sapreexp (Prevent Duplicate Export of Total Values from ReSA)

Module Name	sapreexp.pc
Description	Prevent Duplicate Export of Total Values from ReSA
Functional Area	Oracle Retail Sales Audit
Module Type	Admin
Module Technology	ProC
Integration Catalog ID	RSA20

Design Overview

When a user modifies or revises a transaction through the ReSAuser application, numerous totals may be affected and require re-totaling. The sales audit pre-export module is designed to compare the latest prioritized version of each total defined for export with the version that was previously sent to each system. If they are they same, an SA_EXPORTED entry is created for the total for that particular system, so that the same value will not be exported twice. By determining which totals have not changed since the last export date time (SA_EXPORTED_REV), this module will then create entries on SA_EXPORTED to prohibit any third-party application from receiving multiple export revisions.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This module should be run after the ReSA auditing process and before any export processes.
Pre-Processing	Sarules.pc
Post-Processing	saexpach saexpgl saexpim saexpdw saexpsim saexprms saexpuar
Threading Scheme	NA

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Only two commits will be done. One to establish the store/day lock (this will be done by the package) and one at the end after a store/day or store/day/total has been completely processed.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_EXPORT_LOG	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_TOTAL_USAGE	Yes	No	No	No
SA_EXPORT_LOG	Yes	No	No	No
SA_EXPORTED	Yes	Yes	No	No
SA_EXPORTED_REV	Yes	No	No	No

Integration Contract

Integration Type	NA
File Name	NA
Integration Contract	NA

Design Assumptions

NA

saexprms (Export of POS transactions from ReSA to RMS)

Module Name	saexprms.pc
Description	Export of POS transactions from ReSA to RMS
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA01

Design Overview

The purpose of this batch module is to fetch all sale and return transactions that do not have RMS errors from the ReSA database tables for transmission to Oracle Retail Merchandising System (RMS). Transaction data is rolled up to the item/store/day/price point/sales type level for the SALES transaction type and item/store/day/price point/sales type/no inventory return indicator/return disposition/return warehouse level for the RETURN transaction types.

If the Unit of Work system parameter is defined as S, the whole store/day is skipped if any RMS error is found. If this value is T, only transactions with RMS errors are skipped.

If the transaction has a status of Deleted and it has previously been transmitted, a reversal of the transaction will be sent.

A file is generated for each store/day.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This program should run towards the end of the Sales Auditing cycle where the total (SATOTALS.PC) and rule (SARULES.PC) data are ready to be exported to the external systems.
Pre-Processing	NA
Post-Processing	saprepost saexprms post
Threading Scheme	Multi-threaded by store

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted in batches of pl_commit_max_ctr. Only two commits will be done, one to establish the store/day lock and another at the end, to

release the lock after a store/day has been completely processed. The POSU formatted output file will be created with a temporary name and renamed just before the end of store/day commit.

In case of failure, all work done will be rolled back to the point right after the call to get_lock() and the lock will be released. Thus, the rollback segment should be large enough to hold all inserts into sa_exported for one store/day.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_STORE_DAY	Yes	No	No	No
SA_EXPORT_LOG	Yes	No	Yes	No
V_RESTART_STORE	Yes	No	No	No
STORE	Yes	No	No	No
CURRENCIES	Yes	No	No	No
SA_TRAN_HEAD	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_ERROR_IMPACT	Yes	No	No	No
SA_EXPORTED	Yes	Yes	No	No
SA_TRAN_SEQ_TEMP	No	Yes	No	Yes
SA_TRAN_HEAD_REV	Yes	No	No	No
SA_EXPORTED_REV	Yes	No	No	No
SA_SYSTEM_OPTIONS	Yes	No	No	No
SA_TRAN_ITEM_REV	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
SA_TRAN_DISC_REV	Yes	No	No	No
SA_TRAN_DISC	Yes	No	No	No
SA_TRAN_ITEM	Yes	No	No	No
SA_TRAN_SEQ_TEMP	Yes	Yes	No	Yes
SA_STORE_DAY_READ_LOCK	No	Yes	No	Yes
SA_TRAN_IGTAX_REV	Yes	No	No	No
SA_TRAN_IGTAX	Yes	No	No	No
SA_TRAN_TAX	Yes	No	No	No
SA_TRAN_TAX_REV	Yes	No	No	No
VAT_ITEM	Yes	No	No	No

Integration Contract

Integration Type	Download from ReSA
File Name	POSU_ appended with store number, business date and system date
Integration Contract	IntCon000044

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Identifies the file record type.
	File Line Id	Char(10)	0000000001	Sequential file line number.
	File type definition	Char(4)	POSU	Identifies the file type
	File Create Date	Char(14)		File Create Date in YYYYMMDDHHMMSS format.
	Store	Number(10)		Store location.
	Vat include indicator	Char(1)		Determines whether or not the store values include VAT. Not required, but populated by ReSA.
	Vat region	Number(4)		VAT region the given location is in. Not required, but populated by ReSA.
	Currency code	Char(3)		Currency of the given location. Not required, but populated by ReSA.
	Currency retail decimals	Number(1)		Number of decimals supported by given the currency for retails. Not required, but populated by ReSA.
THEAD	Record descriptor	Char(5)	THEAD	Identifies the file record type.
	File Line Id	Char(10)		Sequential file line number.
	Transaction date	Char(14)		Transaction date in YYYYMMDDHHMMSS format. Corresponds to the date that the sale/return transaction was processed at the POS.
	Item Type	Char(3)	REF or ITM	Can be REF or ITM.
	Item	Char(25)		ID number of the ITM or REF.
	Dept	Number(4)		Department of item sold or returned.
	Class	Number(4)		Class of item sold or returned.

Record Name	Field Name	Field Type	Default Value	Description
	Sub Class	Number(4)		Subclass of item sold or returned.
	Pack Ind	Char(1)		Pack indicator of item sold or returned.
	Item Level	Number(1)		Item level of item sold or returned.
	Tran level	Number(1)		Transaction level of item sold or returned.
	Wastage Type	Char(6)		Wastage type of item sold or returned.
	Wastage pct	Number(12)		Waste pct (4 implied decimal places).
	Tran type	Char(1)		Transaction type code to specify whether transaction is a sale or a return.
	Drop Shipment indicator	Char(1)		Indicates whether the transaction is a drop shipment or not.
	Total sales qty	Number(12)		Total sales quantity (4 implied decimal places).
	Selling UOM	Char(4)		Selling Unit of Measure for the item.
	Sales sign	Char(1)		Determines if the Total Sales Quantity and Total Sales Value are positive or negative.
	Total Sales Value	Number(20)		Total sales value of goods sold/returned (4 implied decimal places).
	Last Date time modified	Char(14)		Date and time of last modification in YYYYMMDDHHMMSS format.
	Catchweight indicator	Char(1)		Indicates if item is a catchweight item.
	Total weight	Number(12)		The actual weight of the item, only populated if catchweight_ind = Y.
	Sub Tran type indicator	Char(1)		Transction type for ReSA. Valid values are A, D, and NULL.
	Total IGTAX Value	Number(20)		This indicates total of all IGTAX amount for the item.

Record Name	Field Name	Field Type	Default Value	Description
	Sales Type	Char(1)		This column indicates whether the line item is a Regular Sale, a customer order serviced by OMS (External CO), or a customer order serviced by a store (In Store CO).
	No Inventory Return Indicator	Char(1)		This column contains an indicator that identifies a return without inventory. This is generally a non-required column, but in the case of Returns, this is required.
	Return Disposition	Char(10)		This column contains the disposition code published by Oracle Retail Warehouse Management Sytem (RWMS0 as part of the Returns upload to OMS.
	Return Warehouse	Char(10)		This column contains the physical warehouse ID for the warehouse identifier where the item was returned.
TTAX	Record descriptor	Char(5)	TTAX	Identifies the file record type.
	File Line Id	Char(10)		Sequential file line number.
	Tax Code	Char(6)		The Tax Code of the item.
	Tax Rate	Number(20)		The tax rate of the item (10 implied decimal places).
	Total Tax Amount	Number(20)		The item level tax or prorated transaction level tax of the item (4 implied decimal places).
TDETL	Record descriptor	Char(5)	TDETL	Identifies the file record type.
	File Line Id	Char(10)		Sequential file line number.
	Promo Tran Type	Char(6)		Code for the promotional type from code_detail where code_type equals PRMT.
	Promotion Number	Number(10)		Promotion number from RMS.
	Sales quantity	Number(12)		Sales quantity sold for this promotion type (4 implied decimal places).
	Sales value	Number(20)		Sales value for this promotion type (4 implied decimal places).

Record Name	Field Name	Field Type	Default Value	Description
	Discount value	Number(20)		Discount value for this promotion type (4 implied decimal places).
	Promotion component	Number(10)		Links the promotion to additional pricing attributes.
TTAIL	Record descriptor	Char(5)	TTAIL	Identifies the file record type.
	File Line Id	Char(10)		Sequential file line number.
	Tran Record Counter	Number(6)		Number of TDETL records in this transaction set.
FTAIL	Record descriptor	Char(5)	FTAIL	Identifies the file record type.
	File Line Id	Number(10)		Sequential file line number.
	File Record counter	Number(10)		Number of records/transactions processed in the current file (only records between head and tail).

Design Assumptions

1. Tax can be sent either in TTAX or IGTAX regardless of yhr default_tax_type of SVAT, GTAX, or SALES. But prorated tax in TTAX will only be sent to RMS in an SVAT configuration since proration is based on VAT_ITEM and VAT_ITEM and is only defined for SVAT.
3. POS can send either transactional level tax details in TTAX lines or item-level tax details in IGTAX lines through the RTLOG file to ReSA. These tax details will be passed on to RMS in the TTAX lines of the POSU file. Even though POS can pass multiple IGTAX/TTAX lines to ReSA and from ReSA to RMS, RMS only supports one tax code per item. If multiple taxes for an item are sent from POS to ReSA, they will be summed to a single tax in RMS sales upload process and assigned one of the applicable tax codes when writing tran_data 88.

saordinvexp (Export Inventory Reservation/Release for In Store Customer Order & Layaway Transactions from ReSA)

Module Name	saordinvexp.pc
Description	Export Inventory Reservation/Release for In Store Customer Order & Layaway Transactions from ReSA
Functional Area	Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA12

Design Overview

This batch program will generate a flat file to reserve or un-reserve the inventory for items on in-store customer order or layaway transactions. Inventory will be reserved for items on customer order/layaway initiate and un-reserved for customer order/layaway cancel or complete transactions.

Customer orders can be categorized into two categories: he In-Store Customer Orders and External Customer Orders. The In-Store Customer Orders are defined as orders that are serviced at the store and inventory reservation is done in Oracle Retail Store Inventory Management (SIM). While the External Customer orders are serviced by an external order management system, no inventory reservation will be made at the store in SIM.

This batch should only process records where the sales type is not equal to External Customer Sales, as it handles only the in-store type orders.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	
Frequency	
Scheduling Considerations	
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	Multithreaded based on store number

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination.

Records are fetched, updated, and inserted in batches of pl_commit_max_ctr. Only two commits are done, one to establish the store/day lock and another at the end, to release the lock after a store/day is completely processed. The ORIN formatted output file is created with a temporary name and renamed just before the end of store/day commit.

In case of failure, all work done is rolled back to the point right after the call to get_lock() and the lock is released. Thus, the rollback segment should be large enough to hold all inserts into sa_exported for one store/day.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_SYSTEM_OPTIONS	Yes	No	No	No
SA_EXPORTED	Yes	Yes	No	No
SA_EXPORT_LOG	Yes	No	Yes	No
SA_STORE_DAY	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_ERROR_IMPACT	Yes	No	No	No
SA_TRAN_HEAD	Yes	No	No	No
SA_TRAN_HEAD_REV	Yes	No	No	No
SA_TRAN_ITEM	Yes	No	No	No
SA_TRAN_ITEM_REV	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No

Integration Contract

Integration Type	Inventory Export from ReSA to RMS
File Name	ORIN_<store>_<tran_date>_<sysdate>
Integration Contract	IntCon000049

Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Identifies the file record type.
	File Line Id	Char(10)	0000000001	Sequential file line number.
	File type Definition	Char(4)	ORIN	Identifies the file type.
	File Create Date	Char(14)		File Create Date in YYYYMMDDHHMMSS format.

Record Name	Field Name	Field Type	Default Value	Description
	Location	Number(10)		Store location number.
THEAD	Record descriptor	Char(5)	THEAD	Identifies the file record type.
	File Line Id	Char(10)		Sequential file line number.
	Transaction Date & Time	Char(14)	Transaction Date	Date and time of the order processed.
	Transaction Type	Char(6)	SALE	Transaction type code specifies whether the transaction is sale or return.
TDETL	Record descriptor	Char(5)	TDETL	Identifies the file record type.
	File Line Id	Char(10)		Sequential file line number.
	Item Type	Char(3)	REF or ITM	Can be REF or ITM.
	Item	Char(25)		ID number of the ITM or REF.
	Item Status	Char(6)	LIN - Layaway Initiate LCA – Layaway Cancel LCO – Layaway Complete PVLCO – Post void of Layaway complete ORI – Pickup/delivery Initiate ORC - Pickup/delivery Cancel ORD - Pickup/delivery Complete PVORD – Post void of Pick-up/delivery complete	Type of transaction.
	Dept	Number(4)		Department of item sold or returned.
	Class	Number(4)		Class of item sold or returned.
	Sub class	Number(4)		Subclass of item sold or returned.
	Pack Ind	Char(1)		Pack indicator of item sold or returned.

Record Name	Field Name	Field Type	Default Value	Description
	Quantity Sign	Chanr(1)	P or N	Sign of the quantity.
	Quantity	Number(12)		Quantity * 10000 (4 implied decimal places), number of units for the given order (item) status.
	Selling UOM	Char(4)		UOM at which this item was sold.
	Catchweight Ind	Char(1)		Indicates if the item is a catchweight item. Valid values are Y or NULL.
	Customer Order number	Char(48)		Customer Order number.
TTAIL	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type.
	File Line Identifier	Number(10)	Specified by ReSA	ID of current line being processed by input file.
	Transaction count	Number(6)	Specified by ReSA	Number of TDETL records in this transaction set.
FTAIL	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type.
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.
	File Record Counter	Number(10)		Number of records/transactions processed in the current file (only records between FHEAD and FTAIL).

Design Assumptions

NA

saexpdw (Export from ReSA to Oracle Retail Analytics)

Module Name	saexpdw.pc
Description	Export from ReSA to Oracle Retail Analytics
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA02

Design Overview

The purpose of this batch module is to fetch all sales and return transactions that do not have Retail Analytics errors from the ReSA database tables for transmission to the Oracle Retail Analytics application. The data will be sent at the store day level. If the transaction has a status of Deleted, and if it has been previously Transmitted, a reversal of the transaction will be sent.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This will run after auditors have made corrections to the data.
Pre-Processing	sapreexp.pc
Post-Processing	resa2dw
Threading Scheme	Multi-threaded by store

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted based on the commit_max_ctr. Only two commits will be done: one to establish the store/day lock and another at the end, to release the lock after a store/day has been completely processed. The RDWT, RDWF, RDWS, and RDWC formatted output files will be created with temporary names and renamed just before the end of store/day commit.

In case of a failure, all the work done will be rolled back to the point right after the call to get_lock() and the lock is released. The rollback segment should be large enough to hold all inserts into sa_exported for one store/day.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_STORE_DAY	Yes	No	No	No
SA_EXPORT_LOG	Yes	No	No	No
V_RESTART_STORE	Yes	No	No	No
SA_STORE_PRICE_HIST_TEMP	No	Yes	No	No
SA_TRAN_HEAD	Yes	No	No	No
SA_CUSTOMER	Yes	No	No	No
SA_STORE_EMP	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_ERROR_IMPACT	Yes	No	No	No
SA_EXPORTED	Yes	No	No	No
SA_TRAN_HEAD_REV	Yes	No	No	No
SA_EXPORTED_REV	Yes	No	No	No
SA_TRAN_ITEM	Yes	No	No	No
SA_TRAN_ITEM_REV	Yes	No	No	No
SA_TRAN_DISC	Yes	No	No	No
SA_TRAN_DISC_REV	Yes	No	No	No
SA_TRAN_TENDER	Yes	No	No	No
SA_VOUCHER	Yes	No	No	No
SA_TRAN_TENDER_REV	Yes	No	No	No
SA_STORE_DAY_READ_LOCK	No	Yes	No	Yes

Integration Contract

Integration Type	Download from ReSA
File Name	RDWT_ appended with store number, business date, and system date. RDWF_ appended with store number, business date, and system date. RDWS_ appended with store number, business date, and system date. RDWC_ appended with store number, business date, and system date.
Integration Contract	IntCon000041 (RDWT) IntCon000156 (RDWF) IntCon000157 (RDWS) IntCon000158 (RDWC)

Four output files will be created for each store_day:

- RDWT - Transaction File
- RDWF - Form of Payment (Tender) file
- RDWS - Store Totals output file
- RDWC - Cashier output File

Each output file is converted into a format for loading into Retail Analytics by the resa2dw Perl script.

Oracle Retail Sales Audit (ReSA) – File Layout – Retail Analytics

File layouts for the interface between sales audit and Retail Analytics.

Char fields are left justified and blank filled.

Number fields are right justified and zero filled. They can contain only numbers.

Numeric fields are left justified and blank filled. They can contain only numbers.

RDWT File

Record Name	Field Name	Field Type	Default Value	Description	Required
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes
	File Type Definition	Char(4)	RDWT	Identifies file as Retail Analytics Transaction file.	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	File Create Date	Number(14)	Create date	Date file was written by external system. Format YYYYMMDDHH24MISS	Yes
Transaction Header	File Type Record Descriptor	Char(5)	THEAD	Identifies transaction record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes
	Business date	Number(8)		Format YYYYMMDD (Note: This is the date the Retail Analytics will consider the transaction date.)	Yes
	Transaction Date	Number(14)	Transaction date	Date sale/return transaction was processed at the POS. Format YYYYMMDDHH24MISS (Note: the Retail Analytics only uses the HH24MI part of this date.)	Yes
	Location	Number(10)	Specified by external system	Store or warehouse identifier.	Yes
	Register ID	Char(5)		The register identifier.	Yes, -1 for null
	Banner ID	Char(4)		The unique identifier of the banner.	Yes, -1 for null
	Line Media ID	Char(10)		The identifier of the media for the order line. For non-merchandise items, such as Shipping & Handling, Service Lines. and gift certificates, the media code will be that of the order line with which it is associated.	Yes, -1 for null
	Selling Item ID	Char(25)		The unique identifier of a selling item.	Yes, -1 for null
	Customer Order Header ID	Char(48)		The unique identifier of a customer order.	Yes, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	Customer Order Line ID	Char(30)		The identifier of a customer order line. For a Value Added Service, such as monogramming, this will be the line number for the item which the service was applied.	Yes, -1 for null
	Customer Order Create Date	Char(8)		The date when the customer order was created/placed.	Yes, -1 for null
	Cashier Identifier	Char(10)		The cashier number. This will be the unique employee number.	Yes, -1 for null
	Salesperson Identifier	Char(10)		The salesperson number. This will be the unique employee number.	Yes, -1 for null
	Customer ID Type	Char(6)		The type of ID number used by this customer.	Yes, -1 for null
	Customer ID Number	Char(16)		Customer ID associated with the transaction.	Yes, -1 for null
	Transaction Number	Number(10)		The unique transaction reference number generated by the POS.	Yes
	Original Register ID	Char(5)		Register ID of the original transaction.	Yes for a transaction type of PVOID.
	Original Transaction Number	Number(10)		Transaction number of the original transaction.	Yes for a transaction type of 'PVOID.
	Transaction Header Number	Numeric(20)		Unique reference used within sales audit to represent the date/store/register/tran_no.	Yes
	Revision number	Number(3)		Number used to identify the version of the transaction being sent.	Yes
	Sales Sign	Char(1)	P - positive N - negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Transaction Type	Char(6)		Transaction type code.	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Sub Transaction Type	Char(6)		The Sub Transaction type.	Yes, -1 for null
	Retail Type	Char(1)	R (Regular), P (Promo), or C (Clearance)		Yes
	Item_Seq_No	Number(4)		The order in which items were entered during the transaction.	No
	Employee Number (Cashier)	Char(10)		Employee identification number. This will only be populated if the sub transaction type is EMP.	Yes, -1 for null
	Receipt Indicator	Char(1)		Flag that identifies returns that have been processed without a receipt. This field will only be populated if the transaction type is RETURN.	No
	Reason Code	Char(6)		A reason is required with a Paid In/Out transaction type, and optional with a return transaction.	Yes, -1 for null
	Vendor number	Numeric(10)		This will only get populated when the paid in code is Expense Vendor.	No
	Item Type	Char(6)	item type identifier	Type of item sold: ITEM, REF, GCN (gift certificate number), or NMITEM.	No
	Item	Char(25)		ID number of the item or gift certificate.	No. Required if Item Type is not null.
	Ref Item	Char(25)		Sub-transaction level item.	No. Also, this field can never be populated without a transaction level item in the item field.
	Taxable Indicator	Char(1)		Taxable/non-taxable status indicator.	No
	Entry/mode	Char(6)		Indicator that identifies whether the item was scanned or manually entered.	No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Department	Number(4)		Department of item sold or returned. Need to validate if using ReSA.	No
	Class	Number(4)		Class of item sold or returned. Need to validate if using ReSA.	No
	Subclass	Number(4)		Subclass of item sold or returned. Need to validate if using ReSA.	No
	Total Sales Quantity	Number(12)		Number of units sold at a particular location, with 4 implied decimal places.	No
	Total Transaction Value	Number(20)		Sales value, net sales value of goods sold/returned, with 4 implied decimal places.	No
	Override Reason	Char(6)		This column will be populated when an item's price has been overridden at the POS to define why it was overridden. This will also always be sent if the transaction originated in RCOM.	Yes, -1 for null
	Return Reason	Char(6)		The reason an item was returned.	Yes, -1 for null
	Total original sign	Char(1)	'P'- positive 'N' – negative		No
	Total Original Sales Value	Number(20)		This column will be populated when the item's price was overridden at the POS and the item's original unit retail is known. This will always be written when the transaction originated in RCOM. This has 4 implied decimals.	No
	Weather	Char(6)		For transaction types of COND, this field will store the type of weather for the store-day.	No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Temperature	Char(6)		For transaction types of COND, this field will store the type of temperature for the store-day.	No
	Traffic	Char(6)		For transaction types of COND, this field will store the type of traffic for the store-day.	No
	Construction	Char(6)		For transaction types of COND, this field will store information regarding any construction on that store-day.	No
	Drop Shipment Indicator	Char(1)	Y or N	Indicates whether the item is involved in a drop shipment.	No
	Item Status	Char(6)		The status of the item, required for voided or exchanged items. Valid values are found in the code_detail table under code_type SASI.	Y, -1 for null
	Tran Process Sys	Char(3)		This column holds the name of the system that processed the transaction. This will be used for filtering duplicate transactions coming from the different systems for export to downstream systems. Expected values are POS – Point of Sale, OMS – Order Management System and, SIM – Store Inventory Management.	Y, -1 for null
	Return Wh	Number(10)		This column contains the physical warehouse ID for the warehouse identifier where the item was returned.	N, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	Fulfill Order No	Char(48)		This column holds the number from OMS related to the fulfillment details. One or more fulfillment orders could relate back to a single customer order in OMS. This column is required if the order is a cross channel order (that is, Sales Type equals E) and the item status is ORD.	N, -1 for null
	No Inventory Return Ind	Char(1)		This column contains an indicator that identifies a return without inventory. This is generally a non-required column, but in the case of returns, this is required.	N
	Sales Type	Char(1)		This column indicates whether the line item is a Regular Sale, a customer order serviced by OMS (External CO), or a customer order serviced by a store (In Store CO).	Y
	Return Disposition	Char(10)		This column will contain the disposition code published by RWMS as part of the Returns upload to OMS.	N, -1 for null
Transaction Detail	File Type Record Descriptor	Char(5)	TDETL	Identifies transaction record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes
	Discount Type	Char(6)		Code for discount type from code_detail, code_type equals SADT.	No
	Promotional Transaction Type	Char(6)		Code for promotional type from code_detail, code_type equals PRMT.	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Promotion Number	Numeric(10)	Promotion number	Promotion number from the RMS.	No
	Promotion Component Number	Numberic(10)	Promo_comp_id from RPM		Required if it is a promotional sale.
	Coupon Number	Char(16)			Yes, if Discount Type is SCOUP.
	Coupon Reference Number	Char(16)			No
	Sales Quantity	Number(12)		Number of units sold in this promotion type, with 4 implied decimal places.	No
	Transaction Sign	Char(1)	P- positive N – negative		Yes
	Transaction Value	Number(20)		Value of units sold in this promotion type, with 4 implied decimal places.	Yes
	Discount Value	Number(20)		Value of discount given in this promotion type, with 4 implied decimal places.	Yes
Transaction Trailer	File Type Record Descriptor	Char(5)	TTAIL	Identifies file record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of current line being processed by input file.	Yes
	Transaction Count	Number(6)	Specified by external system	Number of TDETL records in this transaction set.	Yes
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes
	File Record Counter	Number(10)		Number of records/transactions processed in the current file (only records between head and tail).	Yes

Transaction Item Information Produced by saexpdw.pc after Translation by resa2dw

Record Name	Field Name	Field Type	Default Value	Description	Required
	Business date	Number(8)		Format YYYYMMDD.	Yes
	Transaction Date	Number(14)	Transaction date	Date sale/return transaction was processed at the POS. Format YYYYMMDDHH24MISS	Yes
	Location	Number(10)	Specified by external system	Store or warehouse identifier.	Yes
	Register ID	Char(5)		The register identifier.	Yes, -1 for null
	Banner ID	Char(4)		The unique identifier of the banner.	Yes, -1 for null
	Line Media ID	Char(10)		The identifier of the order line media. For non-merchandise items, such as Shipping & Handling, Service Lines, and gift certificates, the media code will be that of the order line with which is it is associated.	Yes, -1 for null
	Selling Item ID	Char(25)		The unique identifier of a selling item.	Yes, -1 for null
	Customer Order Header ID	Char(48)		The unique identifier of a customer order.	Yes, -1 for null
	Customer Order Line ID	Char(30)		The identifier of a customer order line. For a Value Added Service, such as monogramming, this will be the line number for the item, which the service was applied.	Yes, -1 for null
	Customer Order Create Date	Number(8)		The customer order creation date.	Yes, transaction date for null
	Cashier Identifier	Char(10)		The cashier number. This will be the unique employee number.	Yes, -1 for null
	Salesperson Identifier	Char(10)		The salesperson number. This will be the unique employee number.	Yes, -1 for null
	Customer ID Type	Char(6)		The type of ID number used by this customer.	Yes, -1 for null

Record Name	Field Name	Field Type	Default Value	Description	Required
	Customer ID Number	Char(16)		Customer ID associated with the transaction.	Yes, -1 for null
	Transaction Number	Number(10)		The unique transaction reference number generated by the POS.	Yes
	Original Register ID	Char(5)		Register ID of the original transaction.	Yes for a transaction type of 'PVOID'.
	Original Transaction Number	Number(10)		Transaction number of the original transaction.	Yes for a transaction type of 'PVOID'.
	Transaction Header Number	Numeric(20)		Unique reference used within sales audit to represent the date/store/register/tran_no.	Yes
	Revision number	Number(3)		Number used to identify the version of the transaction being sent.	Yes
	Sales Sign	Char(1)	P - positive N - negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Transaction Type	Char(6)		Transaction type code.	Yes
	Sub Transaction Type	Char(6)		The Sub Transaction type.	Yes, -1 for null
	Retail Type	Char(1)	R (Regular), P (Promo), or C (Clearance)		Yes
	Item_Seq_No	Number(4)		The order in which items were entered during the transaction.	No
	Employee Number (Cashier)	Char(10)		Employee identification number. This will only be populated if the sub transaction type is EMP.	Yes, -1 for null
	Receipt Indicator	Char(1)		Flag that identifies returns that have been processed without a receipt. This field will only be populated if the transaction type is RETURN.	No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Reason Code	Char(6)		A reason is required with a Paid In/Out transaction type, and optional with a return transaction.	Yes, -1 for null
	Vendor number	Numeric(10)		This will only get populated when the paid in code is Expense Vendor.	No
	Item Type	Char(6)	Item type identifier	Type of item sold, ITEM, REF, GCN (gift certificate number), or IMITEM.	No
	Item	Char(25)		ID number of the item or gift certificate.	No. Required if Item Type is not null.
	Ref Item	Char(25)		Sub-transaction level item.	No. Also, this field can never be populated without a transaction level item in the item field.
	Taxable Indicator	Char(1)		Taxable/non-taxable status indicator.	No
	Entry/mode	Char(6)		Indicator that identifies whether the item was scanned or manually entered.	No
	Department	Number(4)		Department of item sold or returned. Need to validate if using ReSA.	No
	Class	Number(4)		Class of item sold or returned. Need to validate if using ReSA.	No
	Subclass	Number(4)		Subclass of item sold or returned. Need to validate if using ReSA.	No
	Total Sales Quantity	Number(12)		Number of units sold at a particular location, with 4 implied decimal places.	No
	Total Transaction Value	Number(20)		Sales value, net sales value of goods sold/returned, with 4 implied decimal places.	No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Override Reason	Char(6)		This column will be populated when an item price has been overridden at the POS to define why it was overridden. This will always be sent if the transaction originated in RCOM.	Yes, -1 for null
	Return Reason	Char(6)		The reason an item was returned.	Yes, -1 for null
	Total original sign	Char(1)	P- positive N - negative		No
	Total Original Sales Value	Number(20)		This column will be populated when the item's price was overridden at the POS and the item's original unit retail is known. This will always be sent if the transaction originated in RCOM. This has 4 implied decimals.	No
	Weather	Char(6)		For transaction types of COND, this field will store the type of weather for the store-day.	No
	Temperature	Char(6)		For transaction types of COND, this field will store the type of temperature for the store-day.	No
	Traffic	Char(6)		For transaction types of COND, this field will store the type of traffic for the store-day.	No
	Construction	Char(6)		For transaction types of COND, this field will store information regarding any construction on that store-day.	No
	Drop Shipment Indicator	Char(1)	Y or N	Indicates whether the item is involved in a drop shipment.	No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Item Status	Char(6)		The status of the item, required for voided or exchanged items. Valid values are found in the code_detail table under code_type SASI.	Y, -1 for null
	Tran Process Sys	Char(3)		This column holds the name of the system that processed the transaction. This will be used for filtering duplicate transactions coming from the different systems for export to downstream systems. Expected values are POS – Point of Sale, OMS – Order Management System and, SIM – Store Inventory Management.	Y, -1 for null
	Return Wh	Number(10)		This column contains the physical warehouse ID for the warehouse identifier where the item was returned.	N, -1 for null
	Fulfill Order No	Char(48)		This column holds the number from OMS related to the fulfillment details. One or more fulfillment orders could relate back to a single customer order in OMS. This column is required if the order is a cross channel order (that is, Sales Type equals E) and the item status is ORD.	N, -1 for null
	No Inventory Return Ind	Char(1)		This column contains an indicator that identifies a return without inventory. This is generally a non-required column, but in case of returns, this is required.	N

Record Name	Field Name	Field Type	Default Value	Description	Required
	Sales Type	Char(1)		This column indicates whether the line item is a Regular Sale, a customer order serviced by OMS (External CO), or a customer order serviced by a store (In Store CO).	Y
	Return Disposition	Char(10)		This column will contain the disposition code published by RWMS as part of the Returns upload to OMS.	N, -1 for null
	Discount Type	Char(6)		Code for discount type from code_detail, code_type equals SADT.	No
	Promotional Transaction Type	Char(6)		Code for promotional type from code_detail, code_type equals PRMT.	Yes
	Promotion Number	Numeric(10)	Promotion number	Promotion number from the RMS.	No
	Promotion Component Number	Numberic(10)	Promo_comp_id from RPM		Required if it is a promotional sale.
	Coupon Number	Char(16)			Yes if Discount Type is SCOUP.
	Coupon Reference Number	Char(16)			No
	Sales Quantity	Number(12)		Number of units sold in this promotion type, with 4 implied decimal places.	No
	Transaction Sign	Char(1)	P - positive N - negative		Yes
	Transaction Value	Number(20)		Value of units sold in this promotion type, with 4 implied decimal places.	Yes
	Discount Value	Number(20)		Value of discount given in this promotion type, with 4 implied decimal places.	Yes

RDWF File

Record Name	Field Name	Field Type	Default Value	Description	Required
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes
	File Type Definition	Char(4)	RDWF	Identifies the file as a Retail Analytics Form of Payment (Tender) file.	Yes
	File Create Date	Numeric(14)	Create date	Date the file was written by external system. Format YYYYMMDDHH24 MISS.	Yes
File Detail	File Type Record Descriptor	Char(5)	FDETL	Identifies file record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes
	Business date	Numeric(8)		Format YYYYMMDD	Yes
	Transaction Date	Numeric(14)	Transaction date	Date sale/return transaction was processed at the POS. Format YYYYMMDDHH24 MISS	Yes
	Location	Number(10)	Specified by external system	Store or warehouse identifier.	Yes
	Cashier Identifier	Char(10)		The cashier number. This will be the unique employee number.	Yes, -1 for null
	Register Identifier	Char(5)			Yes, -1 for null
	Sales Sign	Char(1)	P - positive N - negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Transaction Sequence Number	Numeric(20)		Unique reference used within sales audit to represent the date/store/register /transaction number.	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Revision number	Number(3)		Number used to identify the version of the transaction being sent.	Yes
	Transaction Type	Char(6)		Transaction type code.	Yes
	Tender type group	Char(6)			Yes
	Tender type id	Numeric(6)		Tender type code.	Yes
	Tender amount	Number(20)		Tender amount.	Yes
	Credit Card Number	Numeric(40)			No
	Credit Card Expiration Date	Numeric(8)		Format YYYYMMDD	No
	Credit Card Authorization Number	Char(16)			No
	Credit Card Authorization Source	Char(6)		Contains whether the authorization number was electronically transmitted or manually keyed in after obtaining it through a telephone call. The code type for this field is CCAS.	No
	Credit Card Entry Mode	Char(6)		Contains the method in which the transaction was entered at the POS. Possible entry modes could include: Terminal Used, Magnetic Strip Track One Read, Magnetic Strip Two Read, Magnetic Strip One Transmitted, or Magnetic Strip Two Transmitted. The code type for this field is CCEM.	No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Credit Card Cardholder Verification	Char(6)		Contains the method of identification that was used by the cardholder to verify their identity. Possible values include Signature Verified (S), Card Shown (C), PIN Entered (P), Mail Order / Phone (M). The code type for this field is CCVF.	No
	Credit Card Terminal ID	Char(5)		Contains the identification code of the terminal within the store that the transaction was transmitted.	No
	Credit Card Special Conditions	Char(6)		Contains the special condition of the transaction (mail, phone or electronic-secured or non-secured authentication). The code type for this field is CCSC.	No
	Voucher Number	Char(25)			No
	Voucher Age	Numeric(5)		Age of the gift certificate. Redeemed date minus sold date.	Yes if Tender Type Group is VOUCH.
	Escheat Date	Numeric(8)		Date on which this gift certificate escheats. Format is YYYYMMDD.	Yes if voucher can escheat.
	Coupon Number	Char(16)			Yes if Tender Type Group is COUPON.
	Coupon Reference Number	Char(16)			No. Only if Tender Type Group is COUPON.
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies file record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	File Record Counter	Number(10)		Number of records/transaction processed in the current file (only records between head and tail).	Yes

Retail Analytics Form of Payment File after translation by resa2dw

Record Name	Field Name	Field Type	Default Value	Description	Required
	Business date	Numeric(8)		Format YYYYMMDD	Yes
	Transaction Date	Numeric(14)	Transaction date	Date the sale/return transaction was processed at the POS. Format YYYYMMDDHH24MISS	Yes
	Location	Number(10)	Specified by external system	Store or warehouse identifier.	Yes
	Cashier Identifier	Char(10)		The cashier number. This will be the unique employee number.	Yes, -1 for null
	Register Identifier	Char(5)			Yes, -1 for null
	Sales Sign	Char(1)	P - positive N - negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Transaction Sequence Number	Numeric(20)		Unique reference used within sales audit to represent the date/store/register/transaction number.	Yes
	Revision number	Number(3)		Number used to identify the version of the transaction being sent.	Yes
	Transaction Type	Char(6)		Transaction type code.	Yes
	Tender type group	Char(6)			Yes
	Tender type id	Numeric(6)		Tender type code.	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Tender amount	Number(20)		Tender amount.	Yes
	Credit Card Number	Numeric(40)			No
	Credit Card Expiration Date	Numeric(8)		Format YYYYMMDD	No
	Credit Card Authorization Number	Char(16)			No
	Credit Card Authorization Source	Char(6)		Contains whether the authorization number was electronically transmitted or manually keyed in after obtaining it through a telephone call. The code type for this field is CCAS.	No
	Credit Card Entry Mode	Char(6)		Contains the method in which the transaction was entered at the POS. Possible entry modes could include: Terminal Used, Magnetic Strip Track One Read, Magnetic Strip Two Read, Magnetic Strip One Transmitted, or Magnetic Strip Two Transmitted. The code type for this field is CCEM.	No
	Credit Card Cardholder Verification	Char(6)		Contains the method of identification that was used by the cardholder to verify their identity. Possible values include Signature Verified (S), Card Shown (C), PIN Entered (P), Mail Order / Phone (M). The code type for this field is CCVF.	No

Record Name	Field Name	Field Type	Default Value	Description	Required
	Credit Card Terminal ID	Char(5)		Contains the identification code of the terminal within the store where the transaction was transmitted.	No
	Credit Card Special Conditions	Char(6)		Contains the special condition of the transaction (mail, phone or electronic-secured or non-secured authentication). The code type for this field is CCSC.	No
	Voucher Number	Char(25)			No
	Voucher Age	Numeric(5)		Age of the gift certificate. Redeemed date minus sold date.	Yes if Tender Type Group is VOUCH.
	Escheat Date	Numeric(8)		Date on which this gift certificate escheats. Format is YYYYMMDD.	Yes if voucher can escheat.
	Coupon Number	Char(16)			Yes if Tender Type Group is COUPON.
	Coupon Reference Number	Char(16)			No. Only if Tender Type Group is COUPON.

RDWS File

Record Name	Field Name	Field Type	Default Value	Description	Required
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes
	File Type Definition	Char(4)	RDWS	Identifies file as aRetail Analytics Store Totals file.	Yes
	File Create Date	Numeric(14)	Create date	Date file was written by the external system. Format YYYYMMDDHH24 MISS	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
File Detail	File Type Record Descriptor	Char(5)	FDETL	Identifies the transaction record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes
	Business date	Number(8)		Format YYYYMMDD	Yes
	Location	Number(10)	Specified by external system	Store or warehouse identifier.	Yes
	Sales Sign	Char(1)	P - positive N - negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Total ID	Char(10)		Category identifier used to determine the type of total.	Yes
	Reference Number 1	Char(30)			No
	Reference Number 2	Char(30)			No
	Reference Number 3	Char(30)			No
	Total Sign	Char(1)	P - positive N - negative		Yes
	Total Amount	Number(20)		Total over/short amount, with 4 implied decimal places.	Yes
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies the file record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes
	File Record Counter	Number(10)		Number of records/transactions processed in the current file (only records between head and tail).	Yes

Store Totals Information after Translation by resa2dw

Record Name	Field Name	Field Type	Default Value	Description	Required
	Business date	Number(8)		Format YYYYMMDD	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Location	Number(10)	Specified by external system	Store or warehouse identifier.	Yes
	Sales Sign	Char(1)	P - positive N - negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Total ID	Char(10)		Category identifier used to determine the type of total.	Yes
	Reference Number 1	Char(30)			No
	Reference Number 2	Char(30)			No
	Reference Number 3	Char(30)			No
	Total Sign	Char(1)	P - positive N - negative		Yes
	Total Amount	Number(20)		Total over/short amount, with 4 implied decimal places.	Yes

RDWC File

Record Name	Field Name	Field Type	Default Value	Description	Required
File Header	File Type Record Descriptor	Char(5)	FHEAD	Identifies file record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes
	File Type Definition	Char(4)	RDWC	Identifies the file as a Retail Analytics Cashier/Register Totals file.	Yes
	File Create Date	Numeric(14)	Create date	Date the file was written by the external system. Format YYYYMMDDHH24MISS	Yes
File Detail	File Type Record Descriptor	Char(5)	FDETL	Identifies the transaction record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Business date	Number(8)		Format YYYYMMDD	Yes
	Location	Number(10)	Specified by external system	Store or warehouse identifier.	Yes
	Cashier Identifier	Char(10)		The cashier number.	If Cashier_id is NULL, then Register_id has value. If Cashier_id has value, then Register_id is NULL. Yes, -1 for null
	Register ID	Char(5)		The register identifier.	If Cashier_id is NULL, then Register_id has value. If Cashier_id has value, then Register_id is NULL. Yes, -1 for null
	Sales Sign	Char(1)	P - positive N - negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Total ID	Char(10)		Category identifier used to determine the type of total.	Yes
	Reference Number 1	Char(30)			No
	Reference Number 2	Char(30)			No
	Reference Number 3	Char(30)			No
	Total Sign	Char(1)	P - positive N - negative		Yes
	Total Amount	Number(20)		Total over/short amount, with 4 implied decimal places.	Yes
File Trailer	File Type Record Descriptor	Char(5)	FTAIL	Identifies the file record type.	
	File Line Identifier	Number(10)	Specified by external system	ID of the current line being processed by input file.	Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	File Record Counter	Number(10)		Number of records/transactions processed in the current file (only records between head and tail).	Yes

Cashier/ Register Totals Information after Translation by resa2dw

Record Name	Field Name	Field Type	Default Value	Description	Required
	Business date	Number(8)		Format YYYYMMDD	Yes
	Location	Number(10)	Specified by external system	Store or warehouse identifier	Yes
	Cashier Identifier	Char(10)		The cashier number	If Cashier_id is NULL, then Register_id has value. If Cashier_id has value, then Register_id is NULL. Yes, -1 for null
	Register ID	Char(5)		The register identifier.	If Cashier_id is NULL, then Register_id has value. If Cashier_id has value, then Register_id is NULL. Yes, -1 for null
	Sales Sign	Char(1)	P - positive N - negative	Determines if the Total Sales Quantity and Total Sales Value are positive or negative.	Yes
	Total ID	Char(10)		Category identifier used to determine the type of total.	Yes
	Reference Number 1	Char(30)			No
	Reference Number 2	Char(30)			No
	Reference Number 3	Char(30)			No
	Total Sign	Char(1)	P - positive N - negative		Yes

Record Name	Field Name	Field Type	Default Value	Description	Required
	Total Amount	Number(20)		Total over/short amount, with 4 implied decimal places.	Yes

Design Assumptions

NA

saexpsim (Export of Revised Sale/Return Transactions from ReSA to SIM)

Module Name	Saexpsim.pc
Description	Export of Revised Sale/Return Transactions from ReSA to SIM
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA14

Design Overview

The purpose of this batch module is to fetch all revised sale and return transactions that do not have SIM errors from the ReSA database tables for transmission to SIM. It retrieves all quantity revision transaction data for SALES, RETURN, EEXCH, VOID, and SPLORD transaction types.

If sa_system_options.unit_of_work is S, the whole store/day is skipped if any SIM error is found. If this value is T, then only transactions with SIM errors are skipped.

The batch will only export transactions whose quantity has been revised. The batch will write these revised transactions to the output file along with a reversal of the quantity.

A file of type SIMT is generated for each store/day.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Scheduling Considerations	This program should run towards the end of the Sales Auditing cycle where the total (SATOTALS.PC) and rule (SARULES.PC) data are ready to be exported to the external systems.
Pre-Processing	Satotals, sarules, sapreexp
Post-Processing	saprepost saexpsim post, resa2sim
Threading Scheme	Multi-threaded by store

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted in batches of pl_commit_max_ctr. Only two commits will be done, one to establish the store/day lock and another at the end, to release the lock after a store/day has been completely processed. The SIMT formatted output file will be created with a temporary name and renamed just before the end of store/day commit.

In case of failure, all work done will be rolled back to the point right after the call to get_lock() and the lock released. Thus, the rollback segment should be large enough to hold all inserts into SA_EXPORTED for one store/day.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_STORE_DAY	Yes	No	No	No
SA_EXPORT_LOG	Yes	No	Yes	No
V_RESTART_STORE	Yes	No	No	No
STORE	Yes	No	No	No
SA_TRAN_HEAD	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_ERROR_IMPACT	Yes	No	No	No
SA_EXPORTED	Yes	Yes	No	No
SA_TRAN_HEAD_REV	Yes	No	No	No
SA_EXPORTED_REV	Yes	No	No	No
SA_SYSTEM_OPTIONS	Yes	No	No	No
SA_TRAN_ITEM_REV	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
SA_TRAN_ITEM	Yes	No	No	No
SA_STORE_DAY_READ_LOCK	No	Yes	No	Yes

Integration Contract

Integration Type	Download from ReSA
File Name	SIMT_ appended by store number, business date, and system date
Integration Contract	IntCon000045

Output File

Record Name	Field Name	Field Type	Default Value	Description
FHEAD	Record descriptor	Char(5)	FHEAD	Identifies the file record type.
	File Line Id	Char(10)	0000000001	Sequential file line number.
	File type definition	Char(4)	SIMT	Identifies the file type.
	Store	Number(10)		Store location.

Record Name	Field Name	Field Type	Default Value	Description
	Business Date	Char(8)		Business Date in YYYYMMDD format.
	File Create Date	Char(14)		File Create Date in YYYYMMDDHHMMSS format.
THEAD	Record descriptor	Char(5)	THEAD	Identifies the file record type.
	File Line Id	Char(10)		Sequential file line number.
	Transaction Number	Number(10)		Transaction Identifier.
	Revision Number	Number(3)		Revision Number of the transaction.
	Transaction date	Char(14)		Transaction date in YYYYMMDDHHMMSS format. Corresponds to the date that the transaction occurred.
	Transaction Type	Char(6)		Transaction Type.
	POS Transaction Indicator	Char(1)		Indicates if the transaction was received from POS or manually created. Valid values: Y - POS N - Manual
TDETL	Record descriptor	Char(5)	TDETL	Identifies the file record type.
	File Line Id	Char(10)		Sequential file line number.
	Item Sequence Number	Number(4)		Item sequence number.
	Item	Char(25)		Identifies the merchandise item.
	Item number type	Char(6)		Identifies the type of item number if the item type is ITEM or REF.
	Item Status	Char(6)		Status of the item within the transaction, V for item void, S for sold item, R for returned item. ORI – Order Initiate ORC – Order Cancel ORD – Order Complete LIN – Layaway Initiate LCA – Layaway Cancel LCO – Layaway Complete
	Serial Number	Char(128)		Unique ID.
	Pack Indicator	Char(1)		Pack Indicator.
	Catchweight Indicator	Char(1)		Catchweight Indicator.
	Quantity Sign	Char(1)		Sign of the quantity.

Record Name	Field Name	Field Type	Default Value	Description
	Quantity Value	Number(12)		Number of items, with 4 implied decimal places.
	Standard Unit of Measure	Char(4)		Standard Unit of Measure of the item.
	Selling Unit of Measure	Char(4)		Unit of Measure of the quantity value.
	Waste Type	Char(6)		Waste Type.
	Waste Percent	Number(12)		Waste Percent.
	Drop Ship Indicator	Char(1)		Indicates whether the item is part of a drop shipment.
	Actual Weight	Number(12)		Contains the weight of the item sold, with 4 implied decimal places.
	Actual Weight Sign	Char(1)		Sign of the actual weight.
	Reason Code	Char(6)		Reason entered by the cashier for some transaction types.
	Sales Value	Number(20)		Transaction value, with 4 implied decimal places
	Sales Value Sign	Char(1)		Transaction value sign.
	Unit Retail	Number(20)		Unit retail, with 4 implied decimal places.
	Sales Type	Char(1)		Indicates if the transaction is an In Store Customer Order, External Customer Order, or Regular Sale.
	Customer Order Number	Char(48)		Contains the customer order ID.
	Customer Order Type	Char(6)		Customer order type.
	Fulfillment Order Number	Char(48)		Contains the order ID of the fulfillment order.
TTAIL	Record descriptor	Char(5)	TTAIL	Identifies the file record type.
	File Line Id	Char(10)		Sequential file line number.
	Tran Record Counter	Number(6)		Number of TDETL records in this transaction set.
FTAIL	Record descriptor	Char(5)	FTAIL	Identifies the file record type.
	File Line Id	Number(10)		Sequential file line number.
	File Record counter	Number(10)		Number of records/transactions processed in the current file (only records between head and tail).

Design Assumptions

NA

saexpim (Export DSD and Escheatment from ReSA to Invoice Matching)

Module Name	saexpim.pc
Description	Export DSD and Escheatment from ReSA to Invoice Matching
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA04

Design Overview

The purpose of this program is to support interfacing invoices from Direct Store Delivery and Escheatment sales audit transactions to the Oracle Retail Invoice Matching (ReIM) application. Direct Store Delivery invoices refer to products or services that are delivered to the store and paid for at the store. This program will take DSD invoices that have been staged to the SA_TRAN_HEAD table by the saimptlog.pc program and move them into the INVC_HEAD table. All DSD transactions will be assumed paid. They can be assumed received if there is a proof of delivery number listed on them. Transactions with a vendor invoice ID or a proof of delivery number should be matched to any existing invoice in INVC_HEAD, and that invoice updated with the new information being interfaced. Invoices that do not match an existing invoice in INVC_HEAD will need to be inserted. Each transaction will be exported to INVC_HEAD table only once.

The Sales Audit Transaction type used to identify invoices for Direct Store Delivery transactions will be Paid Out. The Paid Out transaction has a code of PAIDOU. The Sales Audit sub-transaction types will be used to identify whether the invoice is an Expense Vendor Payout or a Merchandise Vendor Payout. The codes are EV for Expense Vendor Payout and MV for Merchandise Vendor Payout. Any Paid Out transaction with a sub-transaction type of Expense Vendor will create a non-merchandise invoice and cause a record to be written to the INVC_NON_MERCH table. ReSA will store non-merchandise codes in the reason_code field on sa_tran_head. Valid values for these reason codes should correspond to the codes stored on the non_merch_code_head table.

In addition to DSD invoices, this program will also interface Escheatment totals to Invoice Matching. Escheatment is the process where an unredeemed gift certificate/voucher or credit voucher will, after a set period of time, be paid out as income to the issuing retailer, or in some states, the state receives this escheatment income. ReSA will be the governing system that determines who receives this income, but Invoice Matching will send the totals, with the related Partner, to an Accounts Payable system. Escheatment information will be stored on the ReSA SA_TOTALS table and will be used to create non-merchandise invoices in Invoice Matching. These invoices will be assumed not paid.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This module should be executed after the ReSA transaction import process after sapreexp. Ideally, after saescheat (if run before, some transactions will not be posted until the next run).
Pre-Processing	Sapreexp, saescheat
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted based on the commit_max_ctr specified on the restart_control table. Only two commits will be done, one to establish the store/day lock and another at the end, to release the lock after a store/day has been completely processed.

In case of failure, all work done will be rolled back to the point right after the call to get_lock and releases the lock. Thus, the rollback segment should be large enough to hold all inserts into sa_exported for one store_day.

Key Tables Affected

Table	Select	Insert	Update	Delete
SYSTEM_OPTIONS	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_EXPORT_LOG	Yes	No	No	No
SA_TRAN_HEAD	Yes	No	No	No
SA_TRAN_TENDER	Yes	No	No	No
SA_EXPORTED	Yes	Yes	No	No
INVC_HEAD	Yes	Yes	Yes	No
INVC_NON_MERCH	No	Yes	Yes	No
INVC_XREF	No	Yes	No	No
TERMS	Yes	No	No	No
SUPS	Yes	No	No	No
PARTNER	Yes	No	No	No
CURRENCY_RATES	Yes	No	No	No
ADDR	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_ERROR_IMPACT	Yes	No	No	No

Integration Contract

Integration Type	Download from ReSA
File Name	NA
Integration Contract	IntCon000043 INVC_HEAD table

Design Assumptions

NA

saexpgl (Post User Defined Totals from ReSA to General Ledger)

Module Name	saexpgl.pc
Description	Post User Defined Totals from ReSA to General Ledger
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA09

Design Overview

The purpose of this module is to post all properly configured user-defined ReSA totals to a general ledger application (Oracle or PeopleSoft). Totals without errors will be posted to the appropriate accounting ledger, as defined in the Sales Audit GL cross-reference module. Depending on the unit of work system parameter, the data will be sent at either the store/day or individual total level. Newly revised totals, that have already been posted to the ledger, will have their previous revision reversed, and the new total posted to the appropriate accounts. Transactions that are from previous periods will be posted to the current period.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This program should run after the ReSA Totaling process (satotals.pc) and Audit Rules process (sarules.pc) and sapreexp.pc.
Pre-Processing	Satotals. Sarules, sapreexp
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted in batches the size of commit max counter. Only one commit will be done after a store/day has been completely processed. A call to release_lock() performs a commit.

Key Tables Affected

Table	Select	Insert	Update	Delete
SYSTEM_OPTIONS	Yes	No	No	No
SA_EXPORT_LOG	Yes	No	Yes	No
STORE	Yes	No	No	No
SA_FIF_GL_CROSS_REF	Yes	No	No	No
STG_FIF_GL_DATA	No	Yes	No	No
IF_ERRORS	No	Yes	No	No
SA_EXPORTED	No	Yes	Yes	No
MV_LOC_SOB	Yes	No	No	No
KEY_MAP_GL	No	Yes	No	No
SA_GL_REF_DATA	No	Yes	No	No
SYSTEM_VARIABLES	Yes	No	No	No

Integration Contract

Integration Type	Download from ReSA
File Name	N/A
Integration Contract	IntCon000019 STG_FIF_GL_DATA

Design Assumptions

NA

ang_saplgen (Extract of POS Transactions by Store/Date from ReSA for Web Search)

Module Name	ang_saplgen.pc
Description	Extract of POS Transactions by Store/Date from ReSA for Web Search
Functional Area	Oracle Retail Sales Audit (ReSA)
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RMS162

Design Overview

The purpose of this batch module is to fetch all corrected sale and return transactions that do not have RMS errors from the ReSA database tables for transmission to an external web search engine. If the transaction has a status of Deleted or Post Voided and has previously been transmitted, a reversal of the transaction will be sent. A file of type POSLOG is generated for each store/day.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This program should run towards the end of the Sales Auditing cycle and before SAEXPRMS.PC.
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	Multi-threaded by store

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, in batches of pl_commit_max_ctr. The POSLOG formatted output file will be created with a completion of store/day looping.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_STORE_DAY	Yes	No	No	No
SA_TRAN_HEAD	Yes	No	No	No
SA_TRAN_ITEM	Yes	No	No	No
SA_EXPORT_LOG	Yes	No	No	No

Table	Select	Insert	Update	Delete
SA_EXPORTED	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_ERROR_IMPACT	Yes	No	No	No
STORE	Yes	No	No	No
ITEM_MASTER	Yes	No	No	No
ITEM_LOC	Yes	No	No	No
DEPS	Yes	No	No	No
CODE_DETAIL	Yes	No	No	No
RPM_ITEM_ZONE_PRICE	Yes	No	No	No
RPM_ZONE_LOCATION	Yes	No	No	No

Integration Contract

Integration Type	Download from ReSA
File Name	POSLOG_<store>_<business date>_<system date>.xml
Integration Contract	IntCon000018

Output File Layout

Record Name	Field Name	Field Type	Default Value	Description
	BatchID	CHAR(18)		A concatenation of store number and business date for a store.
	RetailStoreID	CHAR(10)		The store number for which the POSLog file has to be extracted.
	WorkStationID	CHAR(5)		RegistryID for the store.
	TillID	CHAR(5)		RegistryID for the store.
	SequenceNumber	CHAR(10)		Point of Sale system defined transaction number associated with a transaction.
	BeginDate	CHAR(8)		Starting date time of the transaction.
	EndDate	CHAR(8)		End date time of the transaction.
	CurrencyCode	CHAR(3)		Code of the currency used during the transaction.
	VoidFlag	CHAR(5)		Indicates if the item in the transaction is voided or not. Valid values are TRUE and FALSE.

Record Name	Field Name	Field Type	Default Value	Description
	Item_Status	CHAR(40)		Status of the item is required for voided, exchanged, or returned item.
	MerchandisingHierarchy	CHAR(4)		Department number to which the item belongs.
	Description	CHAR(250)		Item description that has been sold.
	Item	CHAR(25)		Item number.
	TaxIncludedInPrice	CHAR(5)		Indicates if the item is being taxed or not. Valid values are TRUE and FALSE.
	RegularSalesUnitPrice	CHAR(20)		Field holds the unit retail in the standard unit of retail for the item/location combination.
	ActualSalesUnitPrice	CHAR(20)		Retail price for the item.
	ExtendedAmount	CHAR(20)		Total sales for the item in the detail level.
	Qty	CHAR(21)		Unit sold of the item.

Design Assumptions

NA

saescheat (Download of Escheated Vouchers from ReSA for Payment)

Module Name	saescheat.pc
Description	Download of Escheated Vouchers from ReSA for Payment
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA05

Design Overview

The laws of individual states and countries may require a retailer to return monies for aged, unclaimed gift certificates, and vouchers. This process is called escheatment. This program writes records for this data to tables that are read into Oracle Retail Invoice matching (ReIM) by the program saexpim.pc. The data can then be sent as invoices approved for payment to a financial application.

The saescheat batch program will set the status of vouchers that have met certain state's escheats rules or have expired to the proper status and produce a total for later export to Invoice Matching. The rules for escheatment are defined on the sa_escheatment_options table. This program calls the function nextEscheatSeqNo () in saescheat_nextesn batch program, which will select a block of available sequence numbers.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Monthly
Scheduling Considerations	Should run after ReSA Totalling and Auding process (satotals.pc and sarules.pc) and before the export to Invoice Matching (saexpim.pc) and Sales Audit purge (sapurge.pc).
Pre-Processing	N/A
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

The logical unit of work is a store/day. The program commits when the number of store/day records processed has reached the commit_max_ctr.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_STORE_DAY	Yes	No	No	No
SA_VOUCHER	Yes	No	Yes	No
STORE	Yes	No	No	No
ADDR	Yes	No	No	No
SA_VOUCHER_OPTIONS	Yes	No	No	No
SA_ESCHEAT_VOUCHER	No	Yes	No	No
SA_ESCHEAT_TOTAL	No	Yes	No	No
SA_ESCHEAT_OPTIONS	Yes	No	No	No
COMPHEAD	Yes	No	No	No

Integration Contract

Integration Type	Download from ReSA
File Name	N/A
Integration Contract	IntCon000039

Design Assumptions

NA

saescheat_nextesn (Generate Next Sequence for Escheatment Processing)

Module Name	saescheat_nextesn.pc
Description	Generate Next Sequence for Escheatment Processing
Functional Area	Oracle Retail Sales Audit
Module Type	Admin
Module Technology	ProC
Integration Catalog ID	RSA25

Design Overview

This batch program gets the next free sequence for use in the saescheat.pc process. This routine goes and gets a block of numbers when starting, and parcels them out as needed. Once they are all used up, it gets another block and returns a pointer to the string containing the next available number or NULL if an error occurs.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Monthly
Scheduling Considerations	This process is executed as a part of the saescheat.pc processing.
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table	Select	Insert	Update	Delete
ALL_SEQUENCES	Yes	No	No	No
SYSTEM_OPTIONS	Yes	No	No	No

Design Assumptions

NA

saexpach (Download from ReSA to Account Clearing House (ACH) System)

Module Name	saexpash.pc
Description	Download from ReSA to Account Clearing House (ACH) System
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA03

Design Overview

This module will post store/day deposit totals to the SA_STORE_ACH table and bank deposit totals for a given day in a file formatted for export to an ACH (Account Clearing House). The ACH export deviations from the typical Sales Audit export in that store/days must be exported even though errors may have occurred for a given day or store (depending on the unit of work defined), and also, the store/day does not need to be closed for the export to occur. The nature of the ACH process is such that as much money as possible must be sent as soon as possible to the consolidating bank. Any adjustments to the amount sent can be made using the sabnkach screen in the online system.

Deposits for store/days that have not been Fully (F) loaded will not be transferred to the consolidating bank. After they are fully loaded, their deposits will be picked up by the next run of this program.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This module should be run towards the end of the Sales Auditing cycle where the total (SATOTALS.PC) and rule (SARULES.PC) data are ready to be exported to the external systems.
Pre-Processing	SAPREEXP.PC (preprocessing of sales auditing export modules that require totals), SATOTALS.PC and SARULES.PC
Post-Processing	NA
Threading Scheme	NA

Restart/Recovery

This module is in two distinct parts, with two different logical units of work. Thus, restart/recovery has to be implemented so that the first part does not get reprocessed in case the program is being restarted. Details on the implementation follow.

The first driving cursor in this module retrieves a store/day to generate ACH totals. Once the first cursor is complete, the second retrieves bank locations by account numbers.

The first Logical Unit of Work (LUW) is defined as a unique store/day combination. Records will be fetched, using the first driving cursor, in batches of commit_max_ctr, but processed and committed one store/day at a time.

The first driving cursor will fetch all store/days that have been Fully Loaded (F), whose audit status is Audited (A), HQ Errors Pending (H), or Store Errors Pending (S) and that are ready to be exported to ACH. Before processing starts, a write lock is obtained using get_lock (). This driving cursor only fetches store/days with a sa_export_log.status of SAES_R. After a store/day is processed, sa_export_log.status is set to SAES_P so that this store/day will not be selected again if the program is restarted. The commit is performed using retek_force_commit after each store/day has been processed and sa_export_log updated, so as to release the lock.

In case a store/day could not be processed due to locking, the store/day information is placed on a list (called locked store/day list) and the next store/day is processed. This list is kept in memory and is available only during processing. If the store for a store/day obtained from the first driving cursor, is on the locked store/day list, then this store/day cannot be processed. This is the case because there is a data dependency such that data from a particular store/day is dependent on data for the same store but at an earlier date. Thus, if a store/day cannot be processed, then subsequent store/days for the same store cannot be processed either. After the driving cursor returns no more data, the program attempts to process each store/day on the list two more times. If the store/day is still locked, then it is skipped entirely and a message is printed to the error log.

The second LUW is a bank account number. Again, records will be fetched in batches of commit_max_ctr. The second driving cursor cannot retrieve information by the LUW because it is possible for the store's currency to be different from the local bank's currency. In that case, a currency conversion is needed.

For each store/day, the query should retrieve the required ACH transfer. The latter is determined by adding the estimated deposit for the next day, the adjustment to the estimate for the current day, and any manual adjustment to the estimate.

Since a store can be associated with different accounts at different banks, only accounts that are consolidated should be retrieved. Since it is possible for the local bank to be in a different country than the consolidating bank, the currency of the partner should also be fetched.

Since processing is dependent on the type of account at the RDFI, the account type should be fetched by this cursor.

Due to differences in transaction processing in cases when the bank is outside the United States, the partner's country should also be fetched. The results of the query should be sorted by partner country. The results of the query should also be ordered by accounts.

Security Considerations

The fact that this program automates the transfer of funds on behalf of the user makes it a likely target for electronic theft. It must be made clear that the responsibility of electronic protection lies with the users themselves.

Following are some tips and recommendation to users:

- A specific user should be used to run the program. This user would be the only one (or one of a few) who has access to this program.
- The umask for this user should be set up so as to prevent other users from reading/writing its files. This would ensure that when the output file is created, it would not be accessible to other users.
- The appropriate permissions should be set up on the directory, which holds the ACH files. The most restrictive decision would be to not allow any other user to view the contents of the directory.
- The password to this user should be kept confidential.
- A secure means of communication should be implemented for transferring the file from where it has been created to the ACH network. This may be done through encryption, or by copying the file to a disk and trusting the courier to deliver the files intact.
- The ACH network needs to be secure.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_ACH_INFO	Yes	No	No	No
COMPHEAD	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
COMPANY_CLOSED	Yes	No	No	No
COMPANY_CLOSED_EXCEP	Yes	No	No	No
LOCATION_CLOSED	Yes	No	No	No
SA_STORE_ACH	Yes	Yes	Yes	No
SA_BANK_STORE	Yes	No	No	No
SA_EXPORT_LOG	Yes	No	Yes	No
STORE	Yes	No	No	No
PARTNER	Yes	No	No	No
CURRENCY_RATES	Yes	No	No	No
SA_BANK_ACH	Yes	Yes	Yes	No

Integration Contract

Integration Type	Download from ReSA
File Name	ACH_ appended with the consolidating routing number, consolidating account number, and current system date.
Integration Contract	IntCon000040

Output File

Record Name	Field Name	Field Type	Default Value	Description
ACH File Header	Section No.	Number(3)	101	Constant number.
	Console Route No	Number(10)		The routing number of the consolidating bank.
	Sender ID	Char(10)		ID used by the Originator to identify itself.
	Current Date	Char(6)		Vdate in YYMMDD format.
	Day Time	Char(4)		Time of file creation in HH24MM format.
	File Header No.	Number(7)	0094101	Constant number.
	Console Bank Name	Char(23)		Name of the Originating Financial Depository Institution.
	Company Name	Char(23)		The name of the company name.
	Ref Code	Char (8)		Reference code.
ACH CCD Batch Header	Section No.	Number(4)	5225	Constant number.
	Company Name	Char(16)		The name of the company.
	Comp Disc Data	Char(20)	NULL	Any kind of data specific to the company.
	Comp Id	Char(10)		Alphanumeric code to identify the company.
	CCD Header Id	Char(3)	CCD	Constant value.
	Comp Entry Desc	Char(10)	"CONSOL "	A short description from the Originator about the purpose of the entry.
	Tomorrow	Char(6)		Vdate+1 in YYMMDD format.
	Tomorrow	Char(6)		Vdate+1 in YYMMDD format.
	Settle Date	Char(3)	NULL	This is inserted by receiving the ACH Operator.
	Reserved	Number(1)	1	Constant number.
	Odfi Id	Number(8)		8-digit routing number of the ODFI.
	Batch No	Number(7)		Batch number.
ACH CBR Batch Header	Section No.	Number(4)	5225	Constant number.
	Company Name	Char(16)		The name of the company.
	Reserved	Char(3)	FV1	Constant value.
	Exch Rate	Number(15)		Exchange rate for the specified currency.
	Reserved	Char(2)	US	Constant value.
	Comp Id	Char(10)		Alphanumeric code to identify the company

Record Name	Field Name	Field Type	Default Value	Description
	CBR Header Id	Char(3)	CBR	Constant value.
	Comp Entry Desc	Char(10)	"CONSOL "	A short description from the Originator about the purpose of the entry.
	Partner Curr Code	Char(3)		Code identifying the currency the partner uses for business transactions.
	Reserved	Char(3)	USD	Constant value.
	Tomorrow	Char(6)		Vdate+1 in YYMMDD forma.
	Settle Date	Char(3)	NULL	This is inserted by the receiving ACH Operator.
	Reserved	Number(1)	1	Constant number.
	Odfi Id	Number(8)		8-digit routing number of the ODFI.
	Batch No	Number(7)		Batch number.
ACH CCD Entry	Section No.	Number(1)	6	Constant number.
	Trans Code	Char(2)		Code used to identify the type of debit and credit. Value accepted are 27 and 37.
	Routing No	Number(9)		Routing number for the bank account.
	Acct No	Char(17)		Account number of the bank.
	Deposit	Number(10)		The amount involved in the transaction* 10000 (4 implied decimal places).
	Id	Char(15)	Null	Identification number. Optional field containing a number used by the Originator to insert its own number for tracing purposes.
	Store Name	Char(22)		Name of the local store.
	Disc Data	Char(2)	Null	Discretionary data. Any kind of data specific to the transaction.
	Reserved	Number(1)	0	Constant number.
	Trace No	Number(15)		Used to uniquely identify each entry within a batch. The first 8 digits contain the routing number of the ODFI and the other 7 contains a sequence number.
ACH CBR	Section No.	Number(1)	6	Constant number.

Record Name	Field Name	Field Type	Default Value	Description
Entry	Trans Code	Char(2)		Code used to identify the type of debit and credit. Values accepted are 27 and 37.
	Routing No	Number(9)		Routing number for the bank account.
	Acct No	Char(17)		Account number of the bank
	Deposit	Number(10)		The amount involved in the transaction* 10000 (4 implied decimal places).
	Id	Char(15)	Null	Identification number. Optional field containing a number used by the Originator to insert its own number for tracing purposes.
	Store Name	Char(22)		Name of the local store.
	Disc Data	Char(2)	Null	Discretionary data. Any kind of data specific to the transaction.
	Reserved	Number(1)	1	Constant number.
	Trace No	Number(15)		Used to uniquely identify each entry within a batch. The first 8 digits contain the routing number of the ODFI and the other 7 contains a sequence number.
ACH CBR Addendum	Section No.	Number(3)	701	Constant number.
	Payment Info	Char(80)	Null	Payment related information.
	Reserved	Number(4)	0001	Constant number.
	Trace Seq No	Number(7)		Sequence number part of the Trace Number of the entry record to which this addendum is referring.
ACH Batch Control	Section No.	Number(4)	8225	Constant number.
	Batch Line Count	Number(6)		The number of entries and addenda in the batch.
	Hash Count	Number(10)		Sum of the RDFI IDs in the detail records.
	Total Batch Debit	Number(12)		Contains the accumulated debit and debit for the file * 10000 (4 implied decimal places).
	Total Batch Credit	Number(12)		Contains the accumulated credit and credit for the file * 10000 (4 implied decimal places).
	Comp Id	Char(10)		An alphanumeric code identifying the company.

Record Name	Field Name	Field Type	Default Value	Description
	Auth	Char(19)	Null	Message Authentication Code. The first 8 characters represent a code from the Data Encryption Standard (DES) algorithm. The remaining eleven characters are blanks.
	Reserved	Char(6)	Null	Reserved.
	ODFI Id	Number(8)		8-digit routing number of the ODFI.
	Batch No	Number(7)		Batch number.
ACH File Control	Section No.	Number(1)	9	Constant number.
	Batch count	Number(6)		The number of batches sent in the file.
	Block count	Number(6)		The number of physical blocks in the file, including both File Header and File Control Records. This is the ceiling of the number of records divided by the blocking factor, which is 10.
	Entry count	Number(8)		The number of entries and addenda in the file.
	Total hash count	Number(10)		Sum of the Entry Hash fields on the Batch Control Records.
	Total file debit	Number(12)		Contains the accumulated debit and debit for the file * 10000 (4 implied decimal places).
	Total file credit,	Number(12)		Contains the accumulated credit and credit for the file * 10000 (4 implied decimal places).
	Reserved	Char(39)	Null	Reserved.
ACH Completed Block	End string	Char(94)		Mark the end of the file: a string of 94 '9' characters. The number of end lines with a string of 94 '9' characters is identified by the following equation: 10 - mod (number of lines in the file, 10).

Design Assumptions

Oracle Retail assumes that there is only one total to be exported for ACH per store/day.

saexpuar (Export to Universal Account Reconciliation System from ReSA)

Module Name	saexpuar.pc
Description	Export to Universal Account Reconciliation System from ReSA
Functional Area	Oracle Retail Sales Audit
Module Type	Integration
Module Technology	ProC
Integration Catalog ID	RSA06

Design Overview

The SAEXPUAR program is used to select the lottery, bank deposit, money order, and credit card totals and write them to output files for export to an external account clearing house application. For each store day, saexpuar posts specified totals to their appropriate output files.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This program should run after the ReSA Totaling process and Audit Rules process.
Pre-Processing	Satotals, sarules, sapreexp
Post-Processing	N/A
Threading Scheme	N/A

Restart/Recovery

The logical unit of work for this module is defined as a unique store/day combination. Records will be fetched, updated, and inserted in batches of commit_max_ctr. Only two commits will be done. One to establish the store/day lock (this will be done by the package) and the other is done at the end, after a store/day has been completely processed.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_STORE_DAY	Yes	No	No	No
SA_EXPORT_LOG	Yes	No	Yes	No
SA_EXPORTED	No	Yes	Yes	No
SA_EXPORTED_REV	Yes	No	No	No

Table	Select	Insert	Update	Delete
SA_TOTAL	Yes	No	No	No
SA_TOTAL_HEAD	Yes	No	No	No
SA_HQ_VALUE	Yes	No	No	No
SA_STORE_VALUE	Yes	No	No	No
SA_SYS_VALUE	Yes	No	No	No
SA_POS_VALUE	Yes	No	No	No
SA_TOTAL_USAGE	Yes	No	No	No
SA_STORE_DAY_WRITE_LOCK	Yes	No	No	No
SA_STORE_DAY_READ_LOCK	Yes	Yes	No	Yes

Integration Contract

Integration Type	Download from ReSA
File Name	UAR usage type appended with system date.
Integration Contract	IntCon000046

Output File Layout

The output file will contain one line for each store/day detail record in a comma-delimited format. The fields are surrounded by double quotes. For example, a record for store 1000 on May 20, 2001 with an amount of 19.99 will look something like this:

"1", "1000", "1999", "20010520","2","","1","","","","","","","","","","","MN","RET"

Field Name	Field Type	Description
Detail Flag	Char	"1" for detail record.
Store	Number	Store number.
Amount	Number	Total Value * 100 (with 2 implied decimal places).
TranDate	Char	Transaction Date in YYYYMMDD format.
UAR TranCode	Char	Transaction Code. "1" for negative amount, "2" for positive amount.
User Defined Value 1	Char	Ref Number 1 on SA_TOTAL.
User Defined Value 2	Char	Total Seq Number on SA_TOTAL.
User Defined Value 3	Char	Ref Number 2 on SA_TOTAL.
User Defined Value 4	Char	Ref Number 3 on SA_TOTAL.
User Defined Value 5	Char	Not used.
User Defined Value 6	Char	Not used.
User Defined Value 7	Char	Not used.

Field Name	Field Type	Description
User Defined Value 8	Char	Not used.
User Defined Value 9	Char	Not used.
User Defined Value 10	Char	Not used.
State	Char	State.
Account	Char	Total Identification on SA_TOTAL.

Design Assumptions

NA

saprepost (Pre/Post Helper Processes for ReSA Batch Programs)

Module Name	saprepost.pc
Description	Pre/Post Helper Processes for ReSA Batch Programs
Functional Area	Oracle Retail Sales Audit
Module Type	Admin
Module Technology	ProC
Integration Catalog ID	RSA26

Design Overview

The Sales Audit pre/post module 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 Sales Audit program have been processed.

This program will take three parameters: username/password to log in to Oracle, a program before or after which this script must run, and an indicator of whether the script is a pre or post function. It will act as a shell script for running all pre-program and post-program updates and purges.

saprepost contains the following helper functions, which are should be individually scheduled with the related main programs.

Catalog ID	Saprepost Job	Related Main Program
	saprepost saimptlog saimptlogi pre	
	saprepost saimptlog saimptlogi post	
	saprepost sapurge pre	
	saprepost sapurge post	
RSA27	saprepost saexprms post	saexprms
RSA28	saprepost saexpdw post	saexpdw
RSA29	saprepost saordinvexp post	saordinvexp
RSA30	saprepost saexpsfm post	
RSA31	saprepost saexpsim post	saexpsim
RSA32	saprepost (saimptlog) (saimptlogi) pre	Either saimptlog or saimptlogi, depending on which is being run. See the detail design for information about how the two related jobs differ.

Catalog ID	Saprepost Job	Related Main Program
RSA33	saprepost saimptlog saimptlogi post	Either saimptlog or saimptlogi, depending on which is being run. See the detail design for information about how the two related jobs differ.
RSA34	saprepost sapurge pre	Sapurge.pc
RSA35	saprepost sapurge post	Sapurge.pc

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	NA
Pre-Processing	NA
Post-Processing	NA
Threading Scheme	NA

Restart/Recovery

NA

Key Tables Affected

Table	Select	Insert	Update	Index	Delete	Truncate	Trigger
SA_EXPORT_LOG	Yes	Yes	No	No	No	N	No
SA_STORE_DAY	Yes	No	No	No	No	No	No
SA_TRAN_SEQ_TEMP	No	No	No	No	No	Yes	No
ALL_OBJECTS	Yes	No	No	No	No	No	No
ALL_SYNONYMS	Yes	No	No	No	No	No	No
ALL_CONSTRAINTS	Yes	No	No	No	No	No	No
DBA_OBJECTS	Yes	No	No	No	No	No	No
SA_EXPORTED	No	No	No	Yes	No	No	No
RESTART_PROGRAM_STATUS	Yes	No	Yes	No	No	No	No

Integration Contract

Integration Type	NA
File Name	NA
Integration Contract	NA

Design Assumptions

NA

sapurge (Purge Aged Store/Day Transaction, Total Value and Error Data from ReSA)

Module Name	sapurge.pc
Description	Purge Aged Store/Day Transaction, Total Value and Error Data from ReSA
Functional Area	Oracle Retail Sales Audit
Module Type	Admin
Module Technology	ProC
Integration Catalog ID	RSA21

Design Overview

This program will be run daily to control the size of the tables in the sales audit database. Older information will be deleted to ensure optimal performance of the system as a whole.

Different kinds of data need to be kept in the system for different amounts of time. Transactions, all associated transaction details, and Totals calculated or reported for a store day will be deleted when they meet the following criteria:

- The Business Date for those transactions and totals is older than or equal to today's date minus the days_before_purge parameter set up on the sales audit system parameters.
- No locks exist on the store/day.
- One of the two following statements is true for the store/day:
  - Fully loaded, and all errors either corrected or overridden (sa_store_day.audit_status is A (Audited) and sa_store_day.data_status equals F (Fully loaded)). In addition, there are no outstanding exports (records for the store/day in the sa_export_log table where sa_export_log.status equals R (Ready for export)).
  - Never loaded (sa_store_day.audit_status is U (Unaudited) and sa_store_day.data_status equals R (Ready for import)).

Flash Sales data will be deleted when it meets the following criteria:

- Date is two years before today's date minus the days_before_purge parameter set up on the sales audit system parameters.
- Company open and close dates will also need to be kept for two years plus days_before_purge, so that the historical comparisons in flash sales reporting carry the appropriate weight.

Voucher data will be deleted when it meets the following criteria:

- The redeemed date or the escheat date for the specific voucher type is before today's date minus the purge_no_days on sales audit voucher options table for the corresponding voucher type.

The program can also take in a list of store_day_seq_no to delete. For example, the command line could be: sapurge userid/passwd 1000 1001 1002, where 1000, 1001 and 1003 are store_day_seq_nos that the user wants to delete. These must also meet the criteria defined above. If a store_day_seq_no is passed to this program, but does not meet the criteria, an error will be written out to the error log.

An output file will be created to store a record for each store and business date that was purged. The file name must be passed in at the command line as a parameter to sapurge.

Scheduling Constraints

Schedule Information	Description
Processing Cycle	Ad Hoc
Frequency	Daily
Scheduling Considerations	This program should be run as the last program in the ReSA batch flow. It can be run as part of the daily or monthly ReSA schedules.
Pre-Processing	saprepost sapurge pre
Post-Processing	saprepost sapurge post
Threading Scheme	Threaded by store

Restart/Recovery

Restart/recovery is implicit in purge programs. The program only needs to be run again to restart appropriately.

Key Tables Affected

Table	Select	Insert	Update	Delete
SA_STORE_DAY	Yes	No	Yes	Yes
SA_SYSTEM_OPTIONS	Yes	No	No	No
SA_EXPORT_LOG	Yes	No	No	Yes
SA_TRAN_HEAD_REV	Yes	No	No	Yes
SA_TRAN_ITEM_REV	Yes	No	No	Yes
SA_TRAN_HEAD	Yes	No	No	Yes
SA_TRAN_HEAD_TEMP	Yes	Yes	No	Yes
SA_TOTAL	Yes	No	No	Yes
SA_BALANCE_GROUP	Yes	No	No	Yes
SA_ESCHEAT_TOTAL	Yes	No	No	Yes
SA_VOUCHER_OPTIONS	Yes	No	No	No

Table	Select	Insert	Update	Delete
SA_EXPORTED	No	No	No	Yes
SA_EXPORTED_REV	No	No	No	Yes
SA_ERROR_REV	No	No	No	Yes
SA_TRAN_TAX_REV	No	No	No	Yes
SA_TRAN_DISC_REV	No	No	No	Yes
SA_TRAN_TENDER_REV	No	No	No	Yes
SA_TRAN_TAX	No	No	No	Yes
SA_TRAN_DISC	No	No	No	Yes
SA_TRAN_ITEM	No	No	No	Yes
SA_TRAN_TENDER	No	No	No	Yes
SA_CUST_ATTRIB	No	No	No	Yes
SA_CUSTOMER	No	No	No	Yes
SA_COMMENTS	No	No	No	Yes
SA_ERROR	No	No	No	Yes
SA_POS_VALUE	No	No	No	Yes
SA_POS_VALUE_WKSHT	No	No	No	Yes
SA_SYS_VALUE	No	No	No	Yes
SA_SYS_VALUE_WKSHT	No	No	No	Yes
SA_STORE_VALUE	No	No	No	Yes
SA_HQ_VALUE	No	No	No	Yes
SA_ERROR_WKSHT	No	No	No	Yes
SA_MISSING_TRAN	No	No	No	Yes
SA_IMPORT_LOG	No	No	No	Yes
SA_BANK_ACH	No	No	No	Yes
SA_ESCHEAT_VOUCHER	No	No	No	Yes
SA_STORE_DAY_WRITE_LOCK	No	No	No	Yes
SA_FLASH_SALES	No	No	No	Yes
SA_VOUCHER	No	No	No	Yes
SA_STORE_ACH	No	No	No	Yes
KEY_MAP_GL	No	No	No	Yes
SA_GL_REF_DATA	No	No	No	Yes
SA_TRAN_PAYMENT_REV	No	No	No	Yes
SA_TRAN_IGTAX_REV	No	No	No	Yes
SA_TRAN_ITEM_TEMP	Yes	Yes	No	Yes
SA_TRAN_IGTAX	No	No	No	Yes
SA_TRAN_PAYMENT	No	No	No	Yes

Integration Contract

Integration Type	NA
File Name	An optional output file name is passed into the program as a runtime parameter; the output file lists deleted items.
Integration Contract	NA

Design Assumptions

NA

5

In-Context Launching Task Flows In Retail Applications

Retail applications can expose select task flows that you can directly launch. This feature is referred to as an in-context launch in a Retail application. You can launch these task flows directly through specific URLs.

Retail applications provide information about the various task flows that you can in-context launch into, including the URLs and the required parameters.

You can use these URLs in other web pages as links. For example, the URL to a task flow that invokes the Store Day Search Flow in a ReSA Application can be added as a link to dashboard report on a BI server. Since access to the task flows are initiated through URL links, the Retail application, with the requested task flow opened in a UI Shell local area, is shown in a new browser instance of either a window or a tab.

Limitations of an In-Context Launch Via URLs

This section provides details on how the ReSA application is functionally integrated with other systems (including other Oracle Retail systems). The discussion primarily concerns the flow of ReSA-related business data across the enterprise:

- The In-Context launch feature does not detect if there is a window or tab already open for the Retail application. If you click on a link to a Retail application's task flow, such as a report on the BI server, a new browser window or tab opens even though you already have an existing browser window or tab open for that same Retail application.
- If a BI dashboard is added to the Retail application, and if that dashboard has a report that contains links to an in-context launch task flow in the same Retail application, a new browser window or tab will still be opened.

List of In-Context Launchable Task Flows

The following is the list of in-context launchable task flows:

Source	Mandatory parameter	Optional	Sample Url
StoreDay Search	Store / BusinessDay	AutoExecute, AssignedStores DataStatus, OverAllStatus	http://<host>:<port>/ResaPortal/faces/Home?navModelItemId= SearchStoreDayTF? Store =<store Id>& BusinessDay =< BusinessDay >
Storeday Maintenance	StoreSeqNo/ Store / BusinessDate	TabToDisclose	http://<host>:<port>/ ResaPortal /faces/Home?navModelItemId= MaintainStoreDayTF? StoreSeqNo =<Store day sequence No.>& Store =<Store ID> & BusinessDate =< BusinessDate >

Source	Mandatory parameter	Optional	Sample Url
Transaction Maintenance	TransactionSeqNo		http://<host>:<port>/ ResaPortal /faces/Home?navModelItemId= MaintainTransactionTF? TransactionSeqNo=< TransactionSeqNo>
Transaction Search	TransactionSeq/ StoreId / TranBusinessDate	ErrorExists, AutoExecute	http://<host>:<port>/ ResaPortal /faces/Home?navModelItemId= ManageTransactionTF? TransactionSeq =< TransactionSeq >& Store =<Store ID> & TranBusinessDate =< TranBusinessDate >

6

Customizing Retail Applications

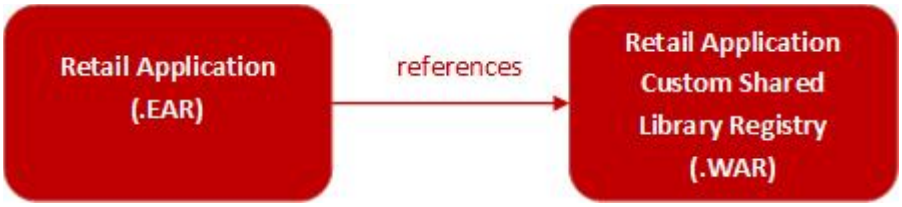
This chapter discusses supported steps for customizing Retail applications.

Using Custom Shared Library for Adding Custom Content

When you want to add new content, such as new pages, business components, or even Java code into a Retail application, you need to create a custom shared library, deploy it into the same managed server as the Retail application, and register that library into the Retail application.

Refer to the WebLogic documentation, Developing Applications with WebLogic Server, to learn more about developing and deploying shared libraries in WebLogic.

A Retail application, that allows for customization, has as part of its installation an intermediary shared library that serves as a registry to reference the actual customer-built shared libraries. The following diagram shows the deployment of a Retail application with the Custom Shared Library Registry.



When you need to add your own content into the shell, metadata to register their content into the shell including the binaries for the content itself (such as, task flows and pages) are expected to be packaged into a Web Archive (WAR) file and deployed as a shared library in the same managed server as the application itself.

Then, the names of these shared libraries have to be referenced in the Custom Shared Library Registry.

Creating and Deploying a Custom Shared Library

This section contains instructions on how you can create your own shared library that will contain custom content they want to include into the Retail Application UI Shell.

The steps in this section are required before you can customize a Retail Application.

Download JDeveloper

To create the custom shared library, it is recommended that you download and install JDeveloper version 11.1.1.7.0 by following the link:

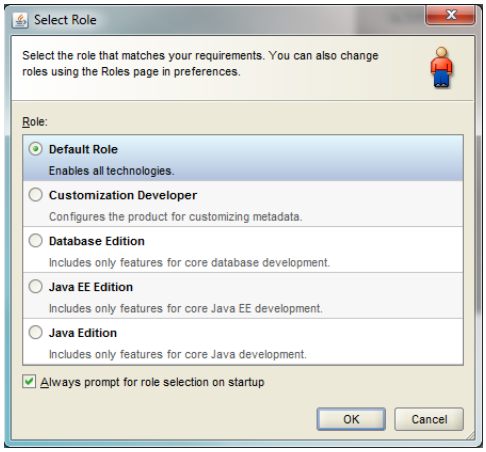
http://www.oracle.com/technetwork/developer-tools/jdev/downloads/index.html

Create the Custom Shared Library Workspace through JDeveloper

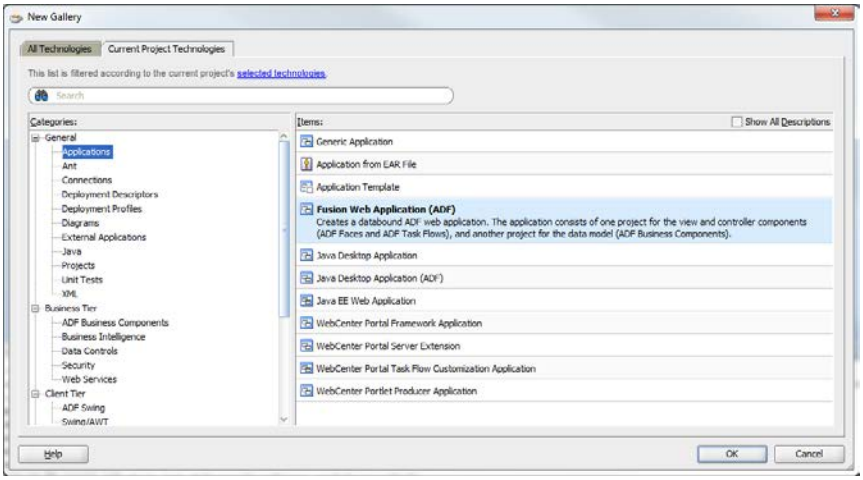
The JDeveloper contains the code for any custom content you want to add to the Retail application. Through JDeveloper, a shared library .WAR file can be generated which can be deployed on the same managed server as the Retail application.

To create and configure the Custom Shared Library Workspace:

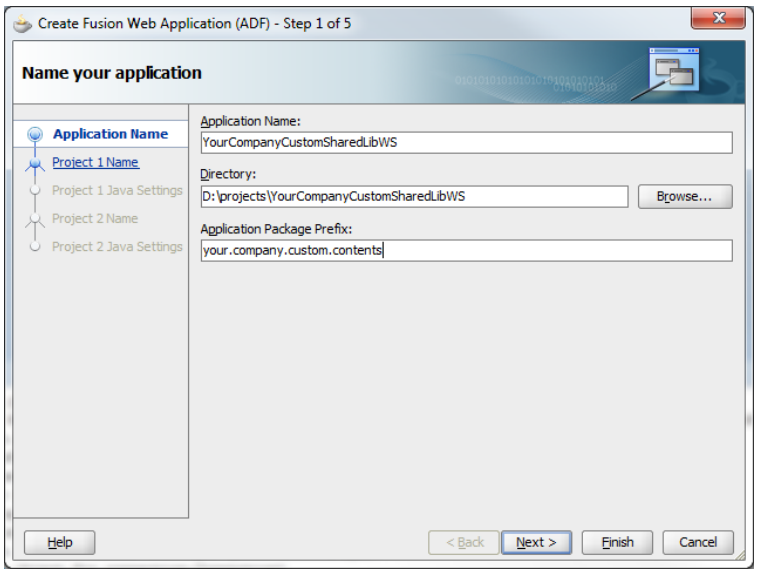
1. Open JDeveloper and choose *Developer Role* when prompted.



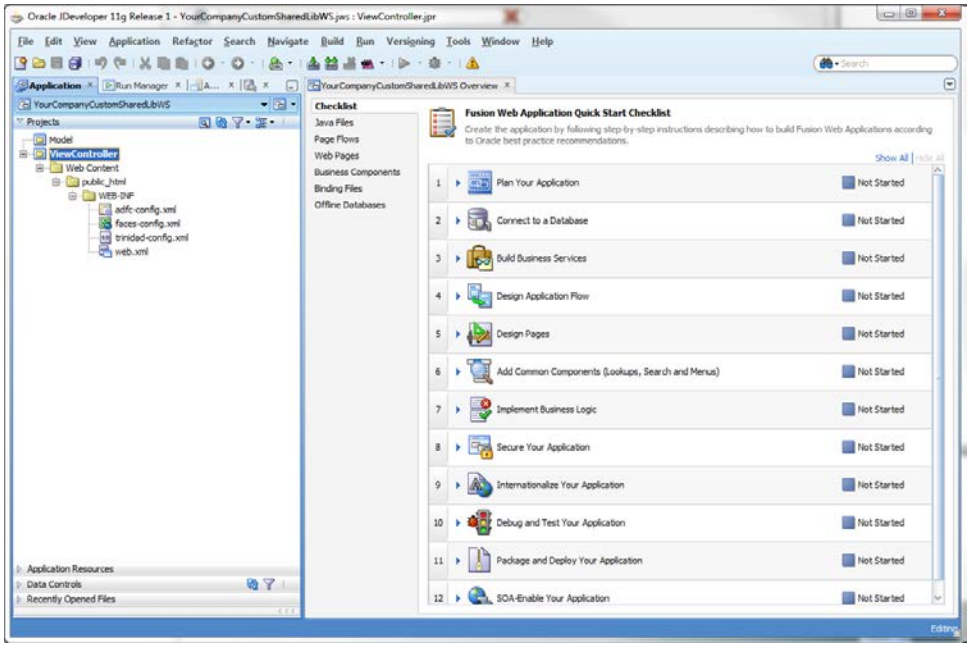
4. Create a new *Fusion Web Application JDeveloper* workspace.
   a. Go to **File > New** to invoke the New Gallery window. Choose **Fusion Web Application (ADF)** as the type of application to create and click **OK**.



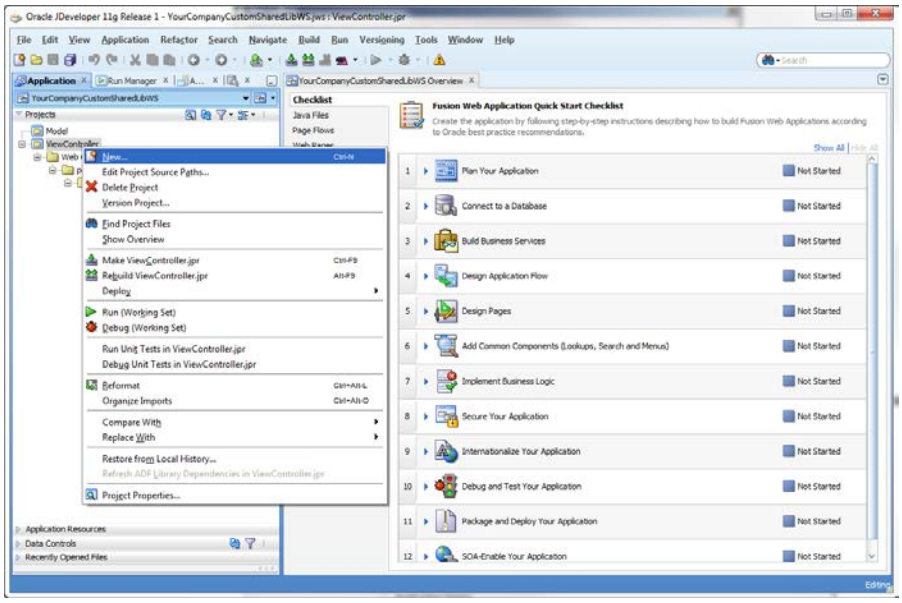
   b. Provide a meaningful application name, a directory path to the workspace, and an application (Java) package prefix. Click **Finish**.



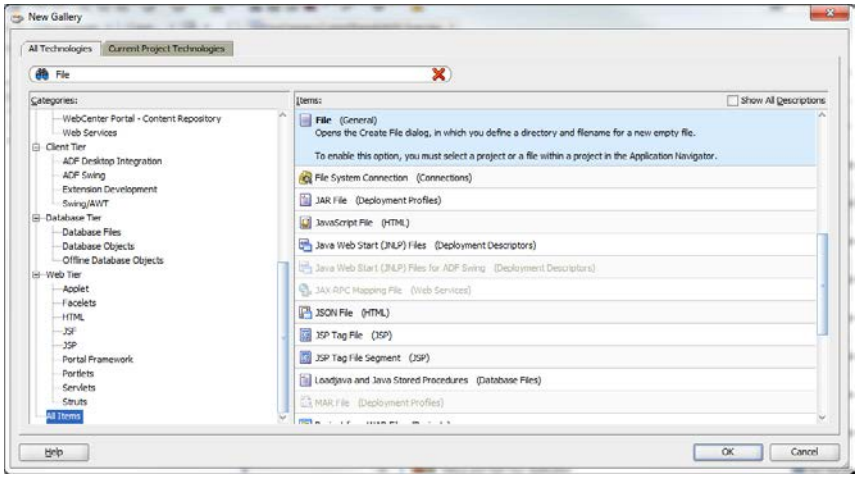
The JDeveloper generates a new workspace with two projects: Model and View-Controller.



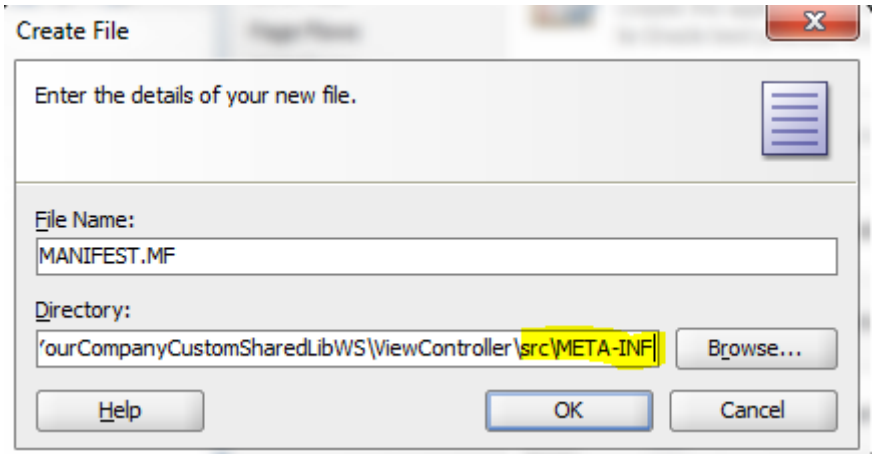
5. Add a Manifest file containing the name of the shared library:
   a. Right-click on the **View-Controller** project and choose **New** from the Context menu.



b. The New Gallery window appears. Choose the option, **File (General)**, in the All Technologies section of the window. Click **OK**.



c. The Create File window opens. For the **File Name**, specify `MANIFEST.MF`. For the **Directory**, the new file must be added under the src/META-INF sub-directory under the View-Controller project's directory.



d. Edit the new `MANIFEST.MF` file and add the following entries:

```
Manifest-Version: 1.0
Implementation-Vendor: companyName
Implementation-Title: Custom Shared Library for companyName
Implementation-Version: 1.0
Extension-Name: companyname.custom.shared.lib
Specification-Version: 1.0
Created-By: companyName
```

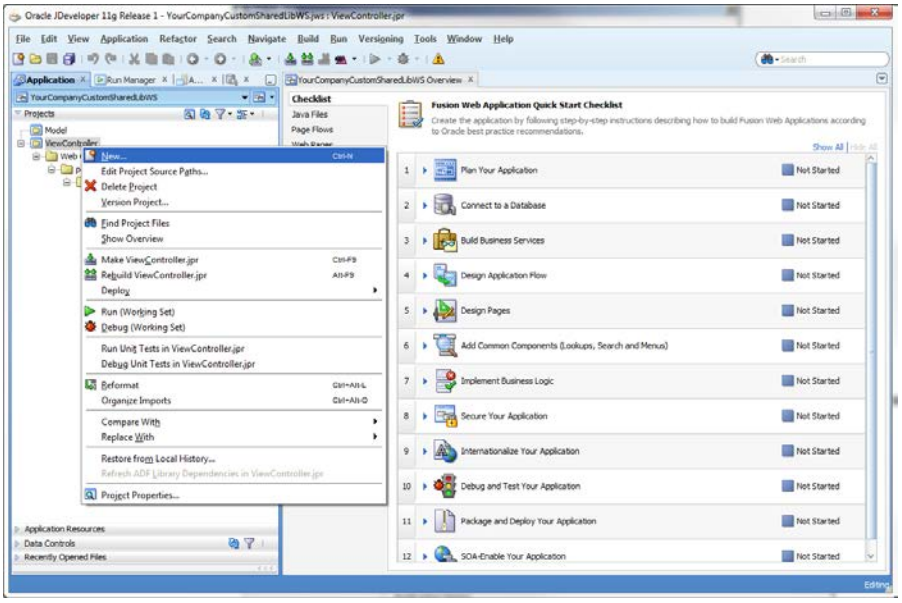
Modify the contents so that meaningful and unique values are used for *Implementation-Vendor*, *Implementation-Title*, *Extension-Name* and *Created-By*.

Example:

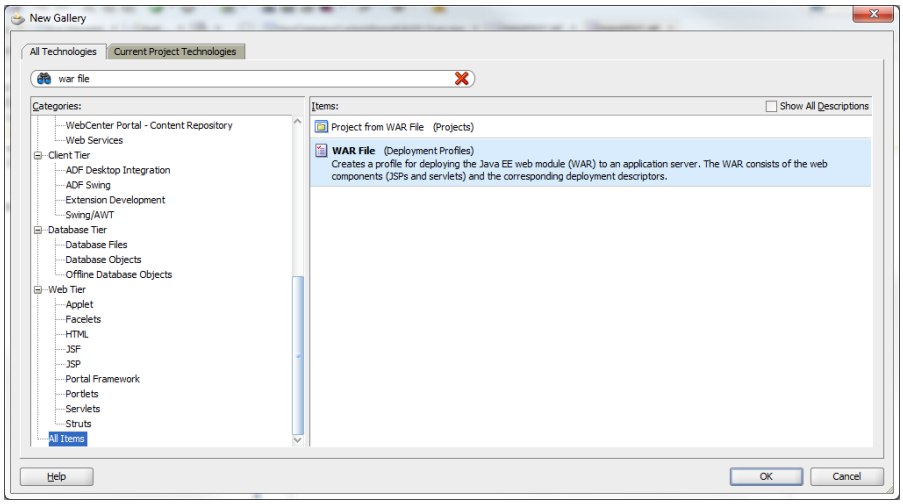
```
Manifest-Version: 1.0
Implementation-Vendor: Acme Retail
Implementation-Title: Custom Shared Library for Acme Retail
Implementation-Version: 1.0
Extension-Name: acmeretail.custom.shared.lib.procurement
Specification-Version: 1.0
Created-By: AcmeRetail
```

6. Create a deployment profile for the shared library.

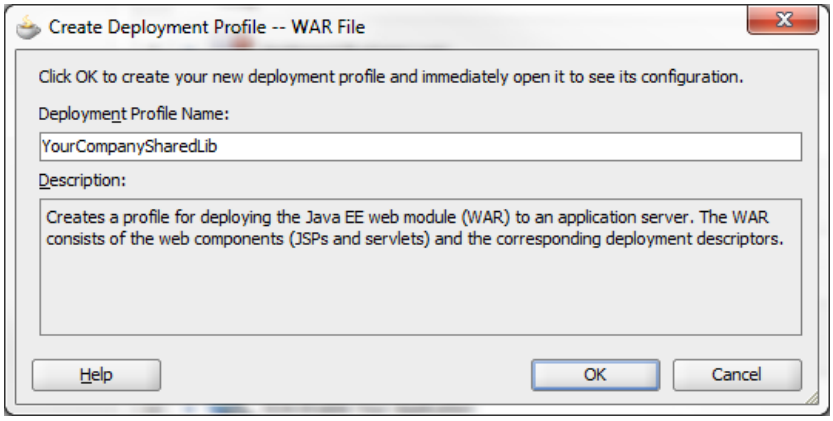
a. Right-click on the **View Controller** project and choose **New** from the context menu.



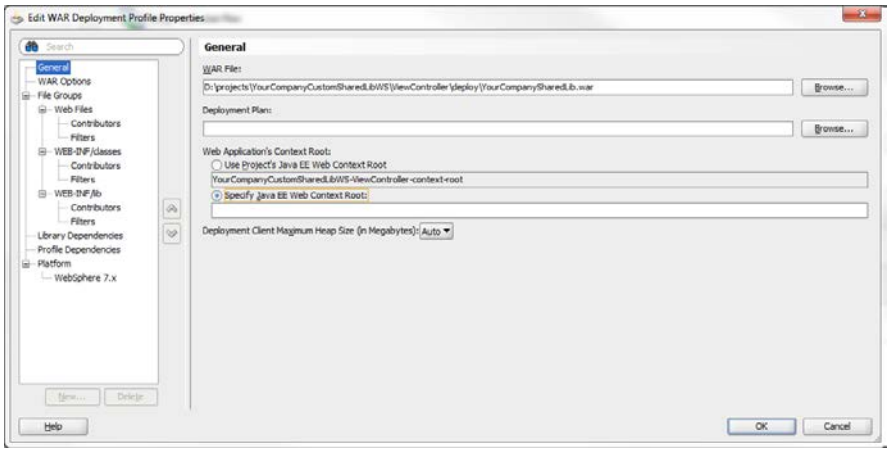
b. The *New Gallery* dialog appears. Choose the option, **WAR File (Deployment Profiles)**, in the All Technologies section of the window. Click **OK**.



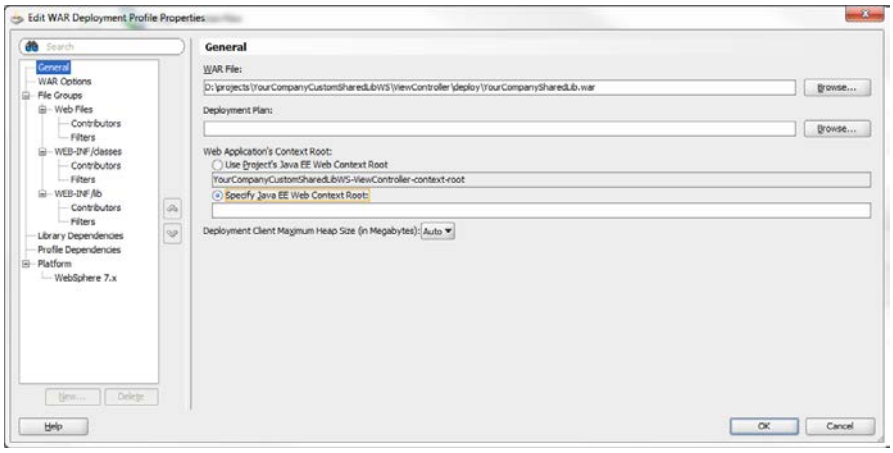
c. Provide a unique and meaningful name for the Deployment Profile and click **OK**.



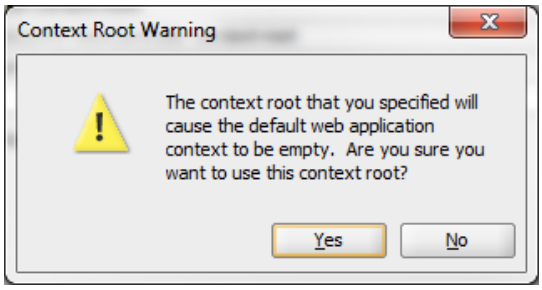
d. The Edit WAR Deployment Profile Properties window appears.



e. Under the General section, make sure that the **Specify Java EE Web Context Root** is selected without any value. Click **OK**.

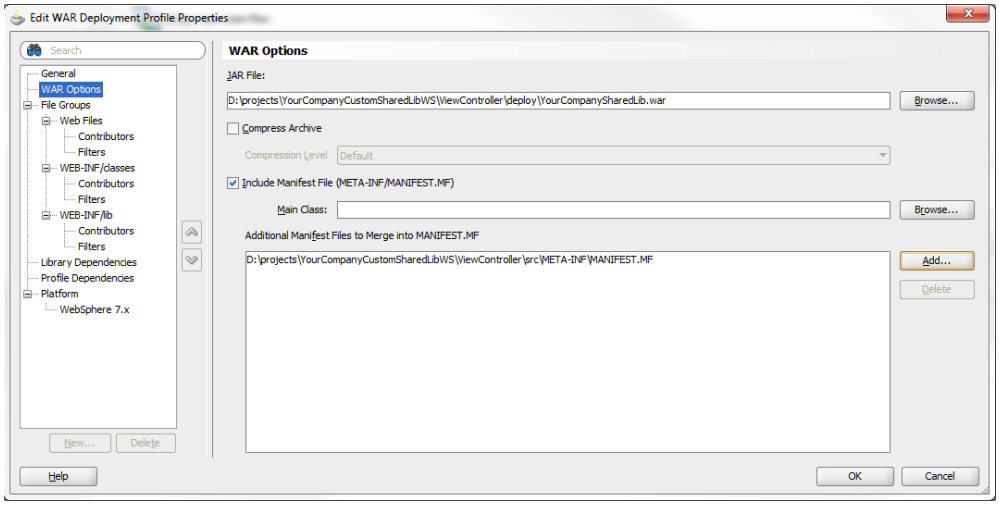


You are prompted to confirm that you really want a blank context root. Click **Yes** to confirm.



f. Under the WAR Options section, enable the **Include Manifest File** option and **Add** the `MANIFEST.MF` file you created under

`.../View-Controller/src/META-INF`.

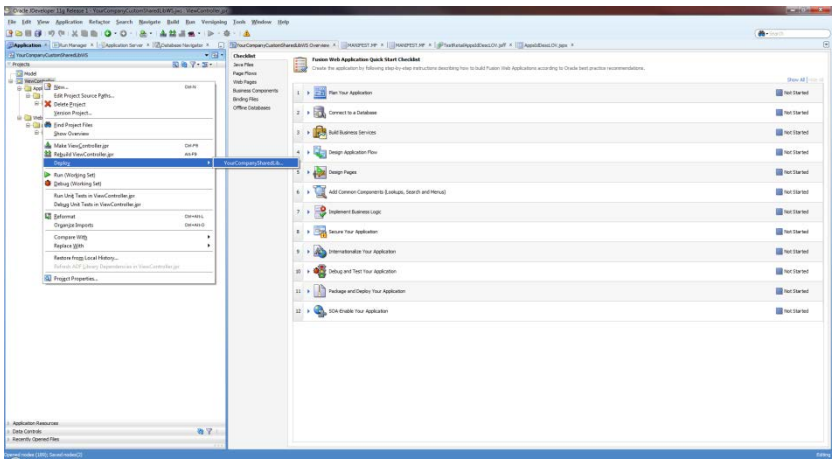


g. Click **OK**.

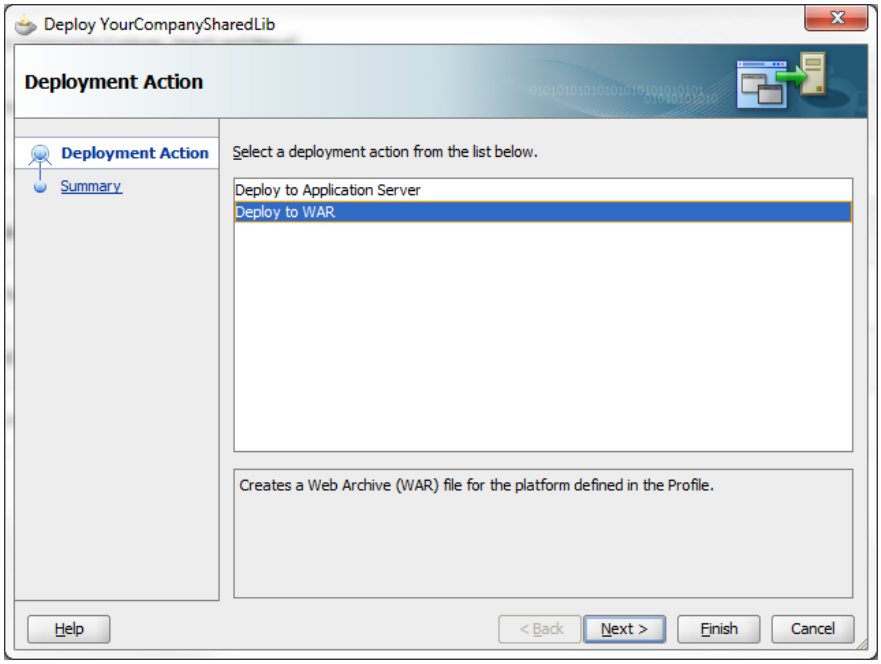
Generate and Deploy the Custom Shared Library WAR

To generate and deploy the Custom Shared Library WAR file, take the following steps:

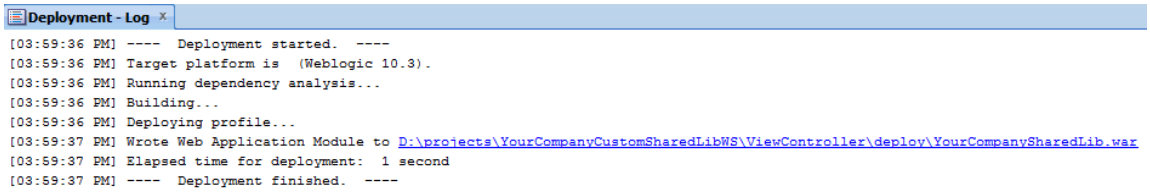
1. Generate the share library WAR file through Jdeveloper:
   a. Open the Custom Shared Library workspace.
   b. Right-click on the **View-Controller** project and choose the shared library WAR deployment profile created previously under the **Deploy** option.



c. The Deploy dialog appears. Select **Deploy to WAR** and click **Finish**



d. The JDeveloper generates the WAR file into the View-Controller project folder sub-directory, deploy.

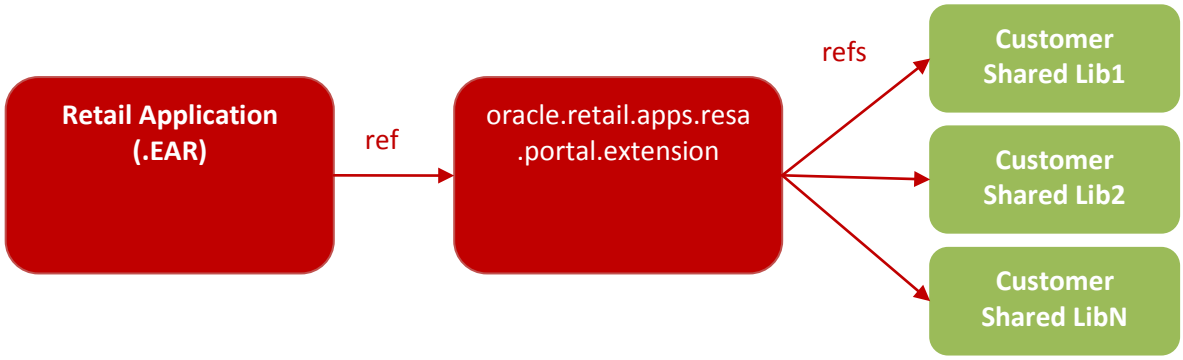


2. Deploy the generated WAR file to the same managed server as the Retail Application as a shared library. For more information, see the section Deploying Applications to Oracle WebLogic Server documentation (http://docs.oracle.com/cd/E23943_01/web.1111/e13702/deploy.htm). This task should be completed using the WebLogic Administration Console or Enterprise Manager Fusion Middleware Control with a user having WebLogic administrator permissions.

Reference the Custom Shared Library from the Retail Application

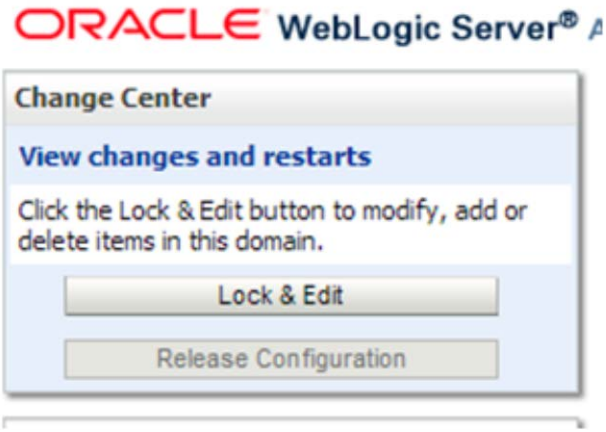
A Retail application, that can be customized, includes in its deployment an intermediary shared library that serves as a registry for any other shared libraries that you want to include during runtime of the Retail application.

Once you create and deploy your own custom shared library by following the steps in this section, you must modify the configuration of the Custom Shared Library Registry to add references to your shared library.

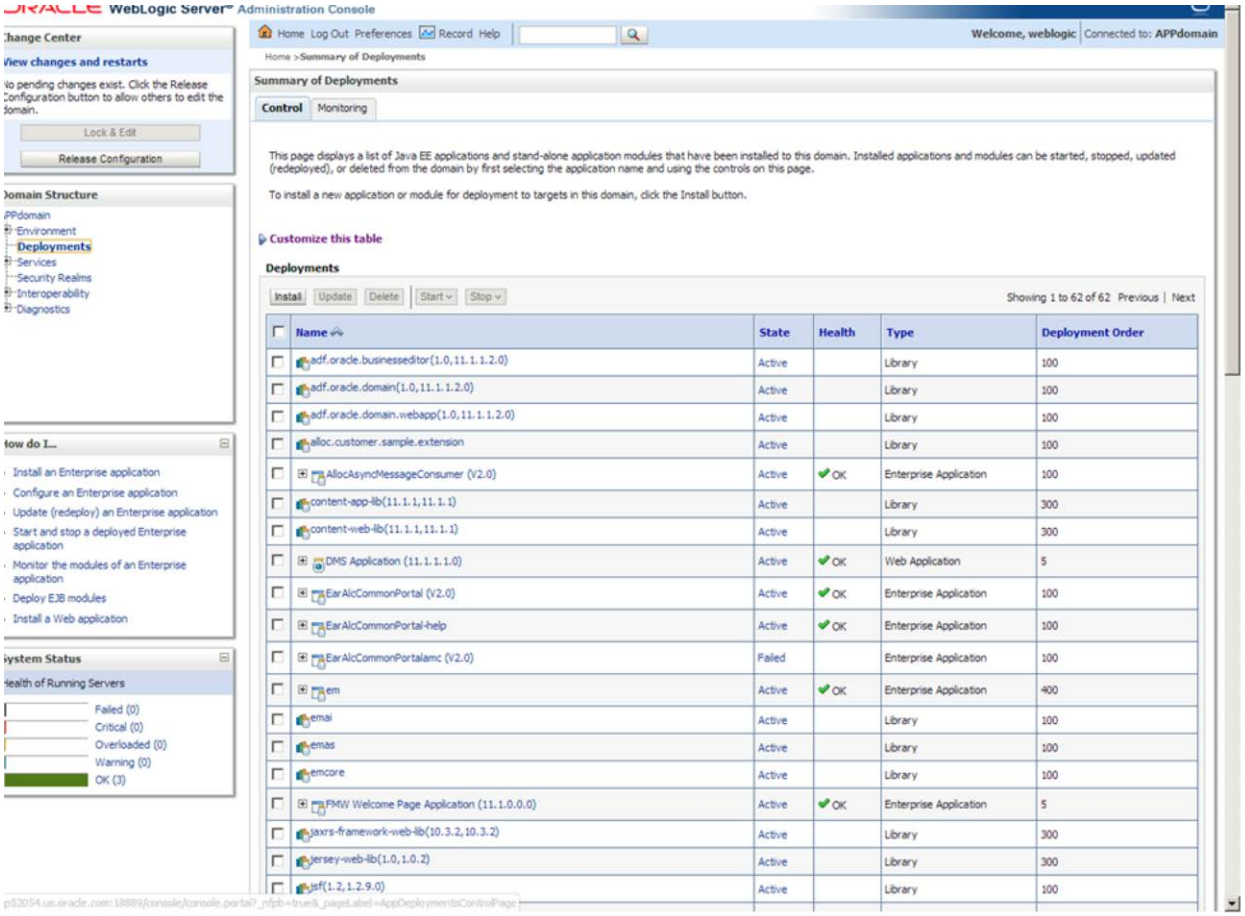


Perform the following steps:

1. Log in to the WebLogic Administration Console as a user with administrative permissions.
2. If the administration console was configured with domain configuration locking, click **Lock & Edit** to ensure that other administrators can be prevented from making changes during your edit session.



3. Navigate to the Deployments section.



4. Look for the Retail Application deployment and shut it down. Choose **Force Stop Now** when appropriate. Wait for the shutdown process to complete.
5. Get the deployment location of the Retail Application's custom shared library registry. Under the deployments list, click on the link for the library named oracle.retail.apps.alc.portal.extensions. The Settings page for the library appears.

   The Settings page will show the file location of the registry's WAR file under the Path entry.

   Make note of this file location.
6. Using the operating system's file manager application, go to the location of the WAR file. You need read and write permissions to the file system where the WAR file is located.
7. Make a copy of the WAR file as back-up.
8. Open the original WAR file using an archive file manager and update the /WEB-INF/`weblogic.xml` by adding a new <library-ref> entry pointing to your custom shared library.

```
<library-ref>
<library-name>companyname.custom.shared.lib</library-name>
</library-ref>
```

> **Note:** The library-name has to match the Extension-Name you provided in your custom shared library's MANIFEST.MF file.

Once this change is done, you have now linked your custom shared library to the Retail Application.

9. Return to the WebLogic Administration Console. Go to Environments' Servers section. Under the Control tab, select the managed server where Retail Application is deployed to. Shut it down and start it back up again.

Adding Custom Content into the Custom Shared Library

The Custom ADF and Java-based components that add new content or functionality into the Retail application can be coded or built into the Custom Shared Library workspace. Typically, you can include task flows that add new UI work flows into the Retail Application UI Shell.

The WAR for the Custom Shared Library needs to be regenerated when new content is added.

To re-deploy the Custom Shared Library WAR, the Retail application and its Custom Shared Library Registry must both be shutdown first.

Since Retail applications are secured web applications, the addition of new ADF UI pages and task flows need to be secured using the application roles and policies that are recognized by the Retail application. The provisioning process is done using the Oracle Enterprise Manager Fusion Middleware Control Console. For more information, see Managing the Policy Store section of the Oracle Fusion Middleware Application Security Guide. (http://docs.oracle.com/cd/E23943_01/core.1111/e10043.pdf).

Customizing the Retail Application UI

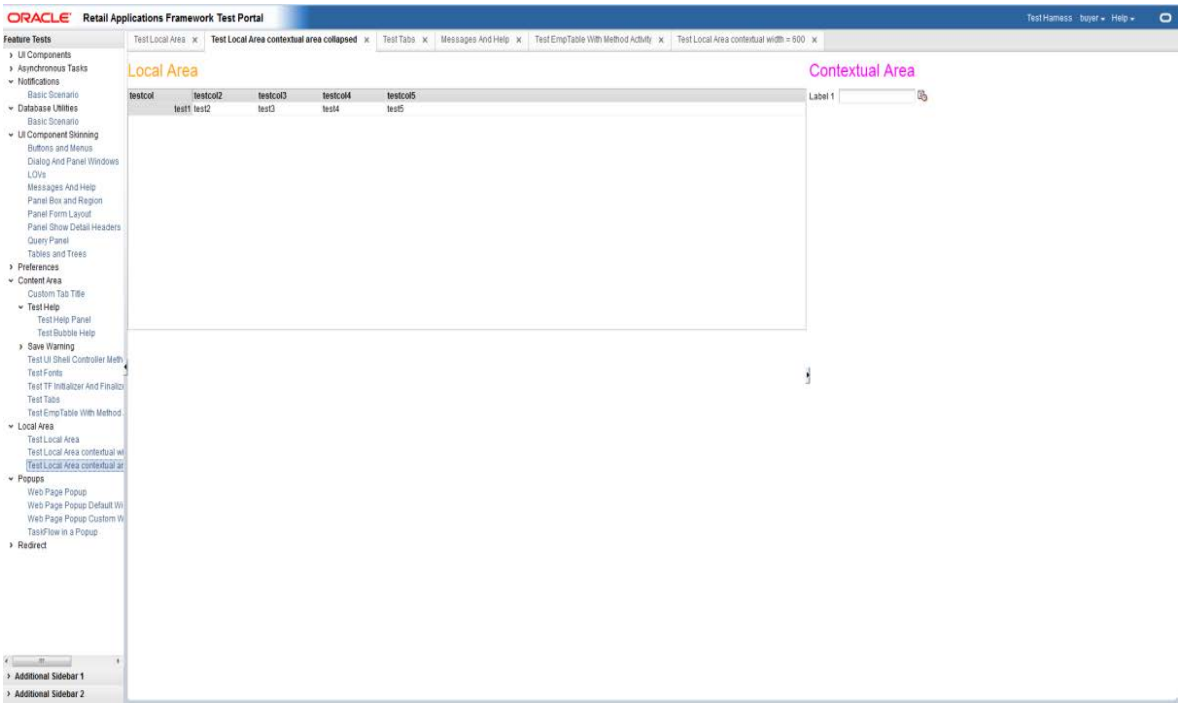
Retail applications can provide you with the capability to add and display custom content such as ADF task flows and URL to BI reports in the deployed application. These custom contents can be accessed from the Retail Application's UI Shell.

Overview of a Retail Application UI Shell

Retail applications are built on top of the Retail Fusion Platform which provide common reusable features that Retail applications can share.

One such feature is the Retail Application UI Shell.

The UI Shell organizes the contents of the application into visual containers that fulfill common layout and navigational requirements in a structured, consistent manner.



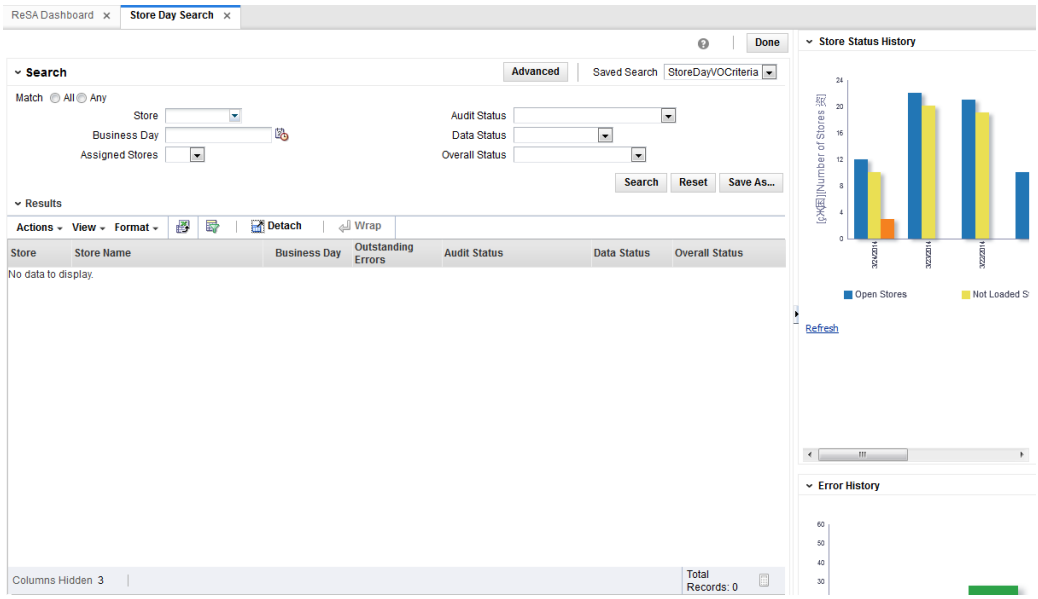
The high-level containers or areas in the shell are highlighted in the diagram below:

The Global Area presents branding and logged-in user information, as well as, menus that allow you to switch to other areas of the application. Notifications are also visually represented as an icon in this area.

The Sidebar Area is a collapsible area on the left of the window. A typical content of the sidebar area is the task navigation tree. Each node on the tree presents a business process or task that opens UI workflow windows in the Local Area.

The Local Area is the main content area for the application. The contents change as a result of actions selected in the Sidebar or Global Area. Typically, task flows are opened as tabs within the Local Area.

The Contextual Area is a collapsible area on the right of the window which provides space to present information that can assist users in completing their tasks. The Contextual Area is presented per the Local Area tab. Each task flow in presented in the local area can have its own contextual area.



Adding a Dashboard

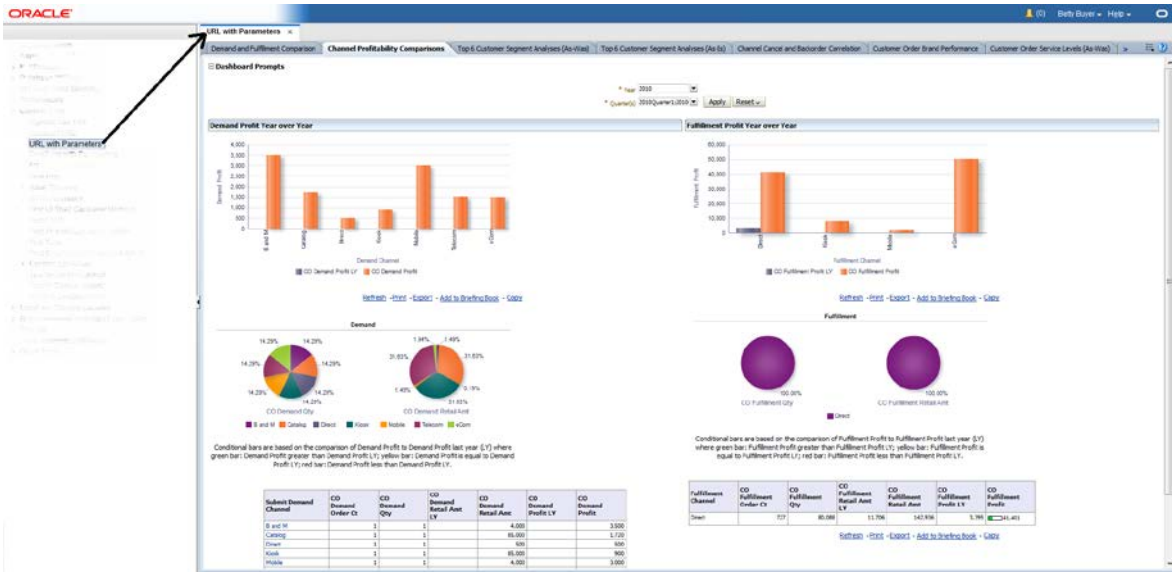
A dashboard is a window within a Retail application that displays the current status of metrics and key performance indicators relevant to the Retail application. They are typically tailored for specific roles and they allow you to easily monitor the status of the current data within the application.

For Retail applications, dashboards are typically built and maintained in a separate BI reporting tool. Oracle Retail recommends using Oracle Business Intelligence Enterprise Edition (OBIEE). Whatever the tool you use, the resulting dashboards must be accessible in a web browser through a URL.

The following is an example dashboard built in OBIEE:

The dashboards can be added into a Retail application which you can launch from the application's UI Shell sidebar task tree.

The dashboard windows are added into a Retail application by adding and configuring the application's sidebar Navigation Model XML file into the Custom Shared Library. The following example displays the dashboard window as rendered in the local area of a Retail application's UI Shell:



Preparing the Custom Shared Library for Adding Dashboards

To prepare the Custom Shared Library to enable you to add dashboard pages into the Retail application, take the following steps:

1. Perform the steps to create a Custom Shared Library workspace, generate a shared library WAR out of it, deploy the WAR, and associate the library to the Retail application.
2. Obtain a copy of the application's sidebar navigation model XML file.

   `EarAllocCore.ear\AlcPublicUIViewController.war\WEB-INF\classes\oracle\retail\apps\framework\uishell\config\custom\HomeSidebarNavigationModel.xml`
3. Using JDeveloper, open the Custom Shared Library workspace in the Developer Role.
4. Add the sidebar model XML file in the View-Controller project src directory, preferably under a sub-directory called Custom.

   For example:

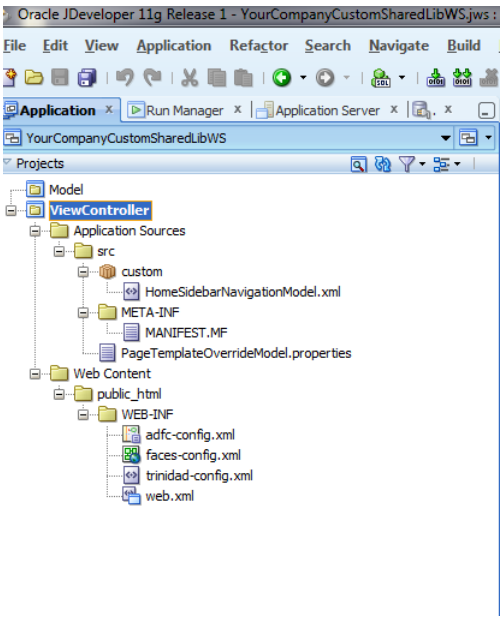
   If the sidebar navigation model is named

   `HomeSidebarNavigationModel.xml`, then that file's path must be `View-Controller/src/custom/HomeSidebarNavigationModel.xml`.
5. Add a new file called PageTemplateOverrideModel.properties under the View-Controller/src directory. Modify this file and add the following entry:

   `Home.sidebarModel=<path to sidebar model xml within view-controller/src>`

   For example:

   `Home.sidebarModel=/custom/HomeSidebarNavigationModel.xml`
6. Verify the location of the files in the workspace.



7. Regenerate the shared library WAR file from the workspace and redeploy the shared library. Make sure you shut down and restart the Retail Application and its shared library registry as a part of this step.
8. Test the Retail Application. The original sidebar contents of the Retail Applications should be accessible.

   The succeeding sections describe the steps to modify the model in order to add dashboard ages into the Retail application.

Adding a Dashboard into a User Interface Shell Sidebar

Adding a dashboard window, so that it is accessible from the sidebar area of the Retail application's UI Shell primary, entails the modification of the Sidebar Navigation Model XML file. The sidebar area renders a collection of links organized or grouped into folders. Each link represents content that launches into a tab in the UI Shell's local content area. A content is typically a workflow that allows you to accomplish specific tasks in the application. The content can be a dashboard page.

Before adding a dashboard into the UI Shell Sidebar, you must have:

- Built, prepared, deployed, and tested the Custom Shared Library as described in the Preparing the Custom Shared Library for Adding Dashboards section.
- Created one or more dashboard pages in their BI reporting tool (such as, OBIEE).
  - The web URL for the dashboard pages must be available in order to proceed with the steps in this section.
  - Any parameters to modify the content of the dashboard must be known and should be accessible as parameters to the dashboard's URL.

Once the preceding pre-requisites have been satisfied, proceed with the following steps:

1. Open the Custom Shared Library workspace in JDeveloper.
2. Open the application's sidebar navigation model XML file (example: `ViewController/src/custom/HomeSidebarNavigationModel.xml`).
3. Go to the bottom of the file and add a new folder into the list. A folder is represented by an `<Item>/<Items>` pair of XML elements of type "`folder`". These have to be added within the topmost `<Items>` tag.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<NavigationDefinition … >
    <Items>
        ... (existing contents)
        <!-- Add this paragraph -->
       <Item id=”myCustomDashboardFolder” title=”My Custom Dashboards”
type=”folder”>
           <Items>
               <!-- add more items here -->
           </Items>
       </Item>
    </Items>
</NavigationDefinition>
```

Provide a unique identifier to the item and meaningful values for the title attributes of the Item tag. For this example, it is assumed that the folder title is "My Custom Dashboards".

9. Add an <Item> element within the folder that references the dashboard page URL as well as the parameters to control the view of the dashboard if available.

   The following example shows an <Item> element that links to a BI dashboard URL with six parameters:.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<NavigationDefinition … >
    <Items>
        ... (existing contents)

       <Item id=”myCustomDashboardsFolder” title=”My Custom Dashboards”
type=”folder”>
           <Items>

               <Item id="myDashboard1"
                     type="link"
                     title="Profitability Dashboard">
                   <url>
<![CDATA[http://mspdv171.us.oracle.com:9704/analytics/saw.dll?SyndicatePag
es&syndicate=portal&PortalPath=/shared/Customer Order/_portal/Customer
Order&Page=Channel Profitability Comparisons&Action=Navigate]]>
                   </url>
                   <Parameters>
                       <Parameter id="col1">"Business Calendar"."Fiscal
Year"</Parameter>
                       <Parameter id="val1">"2010"</Parameter>
                       <Parameter id="psa1">"Retail Merchandising
Analytics As-Was"</Parameter>
                       <Parameter id="col2">"Business Calendar"."Fiscal
Quarter"</Parameter>
                       <Parameter id="val2">"2010Quarter1" "2010Quarter2"
"2010Quarter3"</Parameter>
                       <Parameter id="psa2">"Retail Merchandising
Analytics As-Was"</Parameter>
                   </Parameters>
               </Item>

           </Items>
       </Item>


    </Items>
</NavigationDefinition>
```

To create this element, note the following attributes and sub-elements:

- The <Item> type must be set as "link".
- The <Item> title must be meaningful.
- The <Item> id attribute must be unique across all the other items in the XML file.
- The <url> sub-element within <Item> indicates the URL to the dashboard page built in the BI tool. The entire URL must be marked as character data (that is, enclosed in CDATA).
- The <Parameters> sub-element within <Item> should list all the parameters to the dashboard page if there are any. Each parameter is represented as a <Parameter> element inside <Parameters>:
  - The <Parameter> id should be the actual parameter reference name recognized by the dashboard URL.
  - The value of each <Parameter> is a **string** value. This is the only supported data type.

4. Regenerate the Custom Shared Library WAR file from the Custom Shared Library workspace. Shutdown the Retail application and its Custom Shared Library Registry, redeploy the Custom Shared Library WAR file, and restart the Retail application components.
5. Test the Retail application. Log in and note that a link to the BI dashboard appears in the UI Shell's sidebar task tree under the My Custom Dashboards folder.

Securing Dashboard Access to Specific Roles

To restrict access to the dashboard added on the sidebar to specific security roles, set the visible property on the <Item> element for the dashboard URL to an Expression Language (EL) expression that calls ADF's securityContext API's isUserInRole method. Example:

```
<Item id="myDashboard1"
type="link"
title="Profitability Dashboard"
visible="#{securityContext.isUserInRole['BUYER_JOB']}" >
```

The parameter to the securityContext.isUserInRole method is a logical security role that is configured for the Retail application. The API returns true if the user is included in the specified security role. If the user is not authenticated or is not found in the role, the API returns false.

Adding Contextual Reports

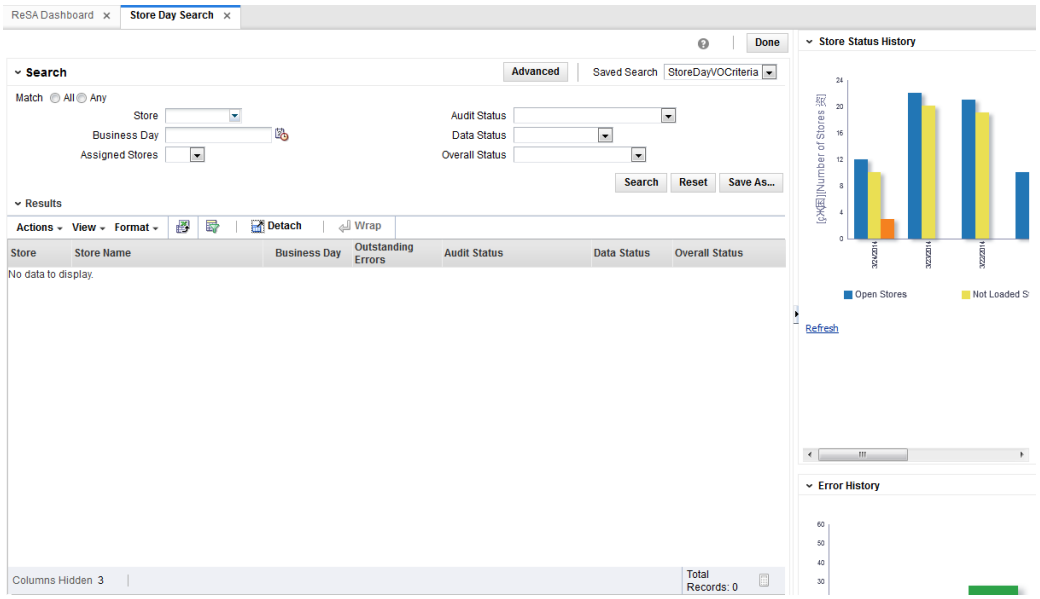
Contextual reports are reports that appear in a task flow's contextual area section. The Contextual Area is a collapsible section to the right of the local area that provides a space to present information that can assist you in completing your tasks.

Since information presented in a contextual area is presented depending on the task or workflow you are on, contextual areas are associated with task flows. There can be at most one contextual area per task flow.

Within a contextual area, multiple contextual reports can be configured.

Each contextual report can change its contents depending on the action being performed in your current workflow.

For example – The count of open store days and not loaded store days for the assigned stores, for the current time period (last week), are shown in the Store Day Search Flow.



Each task flow publishes the contextual business events on key activities taking place on the screen. Contextual reports can listen to those events and change its contents depending on the payload information associated with the event.

Contextual reports are typically built and maintained in a separate BI reporting tool. Oracle Retail recommends using Oracle Business Intelligence Enterprise Edition (OBIEE). Whatever the tool you use, the resulting reports must be accessible in a web browser through a URL.

The contextual reports are added in a Retail application by adding and configuring the task flow's contextual area model XML file into the Custom Shared Library.

Retail applications can also provide you with the list of possible contextual business events each flow generates.You can configure the contextual reports to react to these. Examples of this event include an item being selected, a recalculation being triggered, or a location being selected. Each event includes information about the event's payload information.

Event Name	Task Flow	Page	Contextual Area Model XML Location	Generated when…	Payload Values
UpdateContextAwareReportEvent	SearchStoreDayFlow	SearchStoreday	SearchStoreday_ContextualAreaModel_ContextAwareReport.xml	Launch of Search Store Day screen	▪ xml Lang enableSection508
UpdateContextAwareReportEvent	MaintainStoreDayFlow	MaintainStoreday	MaintainStoreday_ContextualAreaModel_ContextAwareReport.xml	Launch of Store Day Summary screen	▪ Lang ▪ enableSection508 ▪ Store ▪ Store_Day_Seq_No

Event Name	Task Flow	Page	Contextual Area Model XML Location	Generated when…	Payload Values
UpdateContextAwareReportEvent	SearchTenderSummaryFlow	SearchTenderSummary	TenderSummary_ContextualAreaModel_ContextAwareReport.xml	Launch of Tender Summary screen	Lang , enableSection508,Store
UpdateContextAwareReportEvent	MaintainTranHeaderFlow	MaintainTranHeader	MaintainTransaction_ContextualAreaModel_ContextAwareReport.xml	Transaction Maintenance, Item Row change	▪ xml Lang enableSection508 ▪ ReSA Tran. No. ▪ Store ▪ Item

Preparing the Custom Shared Library for Adding Contextual Reports

To prepare the Custom Shared Library to add contextual reports in Retail application task flows:

1. Perform the steps to create a Custom Shared Library workspace, generate a shared library WAR out of it, deploy the WAR, and associate the library to the Retail application.
2. Obtain a copy of the task flow contextual area model XML files where the contextual reports should be added.
3. Using JDeveloper, open the Custom Shared Library workspace in a Developer Role.
4. Add the contextual area model XML file in the **View-Controller** project src directory, preferably under a sub-directory called Custom.

   For example:

   If the contextual area model XML for the task flow, MaintainTranHeaderFlow, is called MaintainTransaction_ContextualAreaModel_ContextAwareReport.xml, that file's path must be `View-Controller/src/custom/` MaintainTransaction_ContextualAreaModel_ContextAwareReport.xml.
5. Add a new, or open the existing file, called `PageTemplateOverrideModel.properties` under the `View-Controller/src` directory. Modify this file and add the following entry:

   ```
   Home.sidebarModel=<path to sidebar model xml within view-controller/src>
   <FlowName>.contextualAreaModel=<path to the contextual area model for the flow>
   ```

   For example:

   ```
   MaintainTranHeaderFlow.sidebarModel=/custom/MaintainTransaction_ContextualAreaModel_ContextAwareReport.xml
   ```
6. Regenerate the shared library WAR file from the workspace and redeploy the shared library. Make sure you shut down and restart the Retail application and its shared library registry as a part of this step.
7. Test the Retail application. Navigate to the flow and make sure the flow is functional.

Adding a Contextual Report To A Task Flow

Adding a contextual report to a task flow primary entails the modification of the task flow's Contextual Area Model XML file. Multiple reports can be added to the model.

Each report is rendered in collapsible panel boxes.

Before adding a dashboard into the UI Shell Sidebar, you must have:

- Built, prepared, deployed, and tested the Custom Shared Library as described in the section.
- Obtained information about the Retail Application's list of contextual business events that can be broadcast from various work flows.
- Created one or more contextual BI reports in the BI reporting tool (such as, OBIEE):
  - The web URL for each report must be available in order to proceed with the steps in this section.
  - Any parameters to configure the content of the report must be known and should be accessible as parameters to the dashboard's URL.

Once the pre-requisites have been satisfied, perform the following steps:

1. Assume the following example scenario when following the steps:
   - A contextual report, called Item Metrics, showing information about an item should be added to the Transaction Maintain Flow's main page. When you select an item on the page, the report displays information for the selected item.
2. Open the Custom Shared Library workspace in JDeveloper.
3. Open the task flow contextual area model XML file (for example, `ViewController/src/custom/` MaintainTransaction_ContextualAreaModel_ContextAwareReport.xml).
4. Add an `<Item>` element within the topmost <Items> element that references the task flow called `ViewContextAwareReportFlow`. The ViewContextAwareReportFlow is a framework for rendering URL-based reports that are aware of contextual business events emanating from the Retail Application task flows.

   For example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<NavigationDefinition … >
    <Items>
        ...
        ...

        <Item id="showItemMetric"
              type="taskflow"
              title="Item Metric">
            <url>
/WEB-
    INF/oracle/retail/apps/framework/contextawarereport/publicui/flow/ViewC
    ontextAwareReportFlow.xml#ViewContextAwareReportFlow
            </url>
         </Item>

    </Items>
</NavigationDefinition>
```

   Note: Note the following:

   - Make sure that the <Item> ID is unique.
   - Make sure the <Item> type is taskflow.

- Provide a meaningful title.

5. Enter the parameters to the `ViewContextAwareReportFlow` by adding the following `<Parameter>/<Parameters>` elements:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<NavigationDefinition … >
    <Items>

        <Item id="showCustomerMetric"
              type="taskflow"
              title=”Customer Metric”>
            <url>
/WEB-
INF/oracle/retail/apps/framework/contextawarereport/publicui/flow/ViewContextA
wareReportFlow.xml#ViewContextAwareReportFlow
            </url>

            <Parameters>
                <Parameter id="reportDescription">Item Metric</Parameter>
                <Parameter
id="actionType">AllocMaintItemSelectedEvent</Parameter>
                <Parameter id="primaryUrl">
<![CDATA[http://companyobiee.com:9704/report/shared/itemMetric&paramItemId=<se
lectedItemId>&paramItemType=<selectedItemType:token01>&paramLanguage=<language
>]]>
                </Parameter>
                <Parameter id="token01">regular</Parameter>
             </Parameters>

         </Item>

    </Items>
</NavigationDefinition>
```

Note: Note the following:

- The `<Parameter id="reportDescription">` element is the title of the contextual area report. Set this to a meaningful value.
- The `<Parameter id="actionType">` element indicates the contextual business event name the report listens to.
- The `<Parameter id="primaryUrl">` element indicates the URL to the contextual area report in the BI server. The entire URL must be marked as character data (that is, enclosed in CDATA). Note how the parameters to the URL are *tokenized*:
    - The `"?paramItemId=<selectedItemId>"` portion of the URL instructs the system to pass the contextual business event payload value called `selectedItemId` into the URL parameter `paramItemId` when rendering the contextual report.
    - The `"?paramItemType=<selectedItemType:token01>"` portion of the URL instructs the system to pass the contextual business event payload value called `selectedItemType` into the URL parameter `paramItemType` when rendering the contextual report. If that payload value is empty or null at runtime, then a default value of `regular` is used as referenced in a `<Parameter id="token01">` entry.
    - The `"?paramLanguage=<language>"` portion of the URL instructs the system about the current locale of the user. The "`language`" identifier is a reference to a value in the contextual event payload. This is a built-in value that all Retail Application contextual business event payloads will have.
- The `<Parameter id="token01">` element holds the default value for the URL parameter selectedItemType. Token parameters hold default values. You can define up to 20 default value tokens.

6. Regenerate the Custom Shared Library WAR file from the Custom Shared Library workspace. Shut down the Retail application and its Custom Shared Library Registry, redeploy the Custom Shared Library WAR file, and restart the Retail application components.
7. Test the Retail application. Go to the flow where the report was added and verify that the report is rendered correctly.

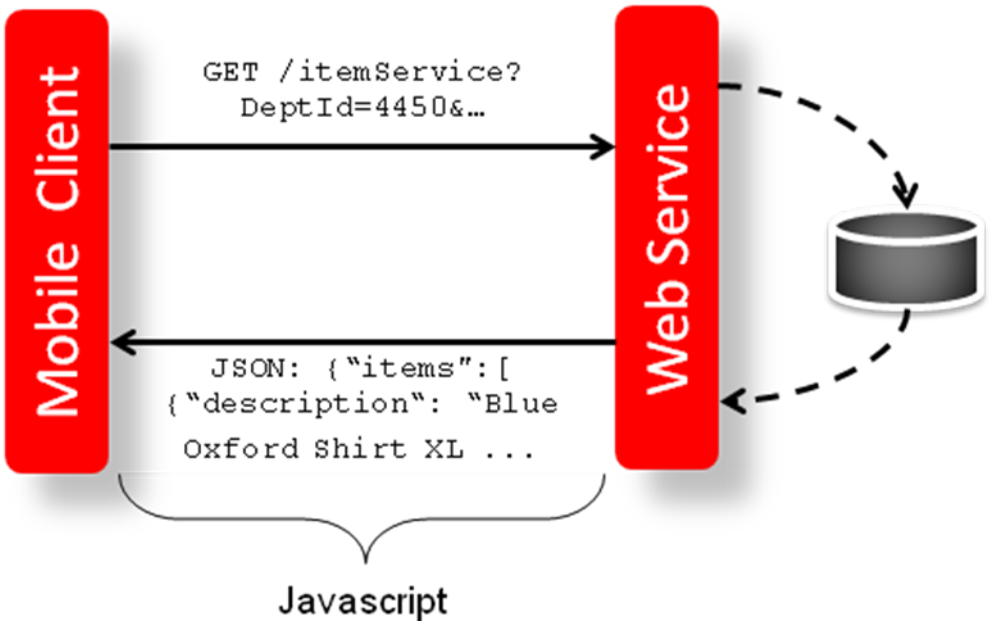
7

ReSA ReSTful Web Service Implementation

This chapter gives an overview about the ReSA ReSTful Web Service Implementation API designs used in the ReSA environment and various functional attributes used in the APIs.

Retailers can access backend functionality in Retail Applications by calling the applications' ReSTful Web Services.

Refer to http://www.oracle.com/technetwork/articles/javase/index-137171.html to learn more about ReST as an architectural style applied to building web services.



The services can be used as is for the client's custom mobile application. The ReSTful Web Services Java code cannot be customized, but the MBL PL/SQL functions and object types can be customized for client use. In addition the client can use the RSL tool to create their own custom services in place of the ReSTful Web Services.

Using ReSTful Web Service during batch window

The services should not be used during the restricted batch window.

Common Characteristics of Retail Application ReSTful Web Services

Deployment

A Retail Application will package its ReST services as part of the application's Enterprise Archive (EAR) file. Specifically, those services are packaged as a Web Archive (WAR) within the EAR.

Installation of the ReST web services is therefore done by default.

Security

Services are secured using J2EE-based security model.

- Realm-based User Authentication. This verifies users through an underlying realm. The user name and password is passed using Http Basic authentication.
- Role-based Authorization. This assigns users to roles, which in turn are granted or restricted access to resources/services. The authorization of ReSTful web services is static and cannot be reassigned to other rules post-installation.
- The communication between the server and client is encrypted using one way SSL.
- RMS\ReSA user data security are implemented in the APIs.

Standard Request and Response Headers

Retail Application ReSTful web services have the following standard HTTP headers:

```
Accept: application/xml or application/JSON
Accept-Version: 14.1 (service version number)
Accept-Language: en-US,en;q=0.8
```

Depending on the type of the operation or HTTP method, the corresponding response header is updated in the Http response with the following codes:

- `GET/READ : 200`
- `PUT/CREATE : 201 created`
- `POST/UPDATE : 204`
- `DELETE : 204`

Standard Error Response

Example response payload in case of service error is depicted below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<messagesRDOes>
  <messagesRDO>
     <message>REST Service Version Mismatch</message>
     <messageType>ERROR</messageType>
     <status>BAD_REQUEST</status>
 </messagesRDO>
</messagesRDOes>
```

- message: The error message - translated.
- messageType: Value of ERROR is returned.
- status : For a bad request or error, the status is BAD_REQUEST.
- The http error code for an error response is 400

URL Path

To access the ReSA ReSTful web services javadoc:

http://host:port/ResaReSTService

To access the ReSA ReSTful web services javadoc:

http://host:port/ResaReSTService/services/private/Resa</service>

HTTP Header

Accept-Version: the value should match the ReSA release version. Example: 14.1

Accept and **Content-Type**: determines output/input style. Supported styles are JSON and XML. Example: application/json

Date Format

All output date fields are in long format, and all input date must also be in long format.

Paging

Some of the ReSA ReSTful web services have the potential to bring back a large number of records and, therefore, these services are equipped to segment the result into pages. The page number to retrieve and size of the page are added as input parameters to all the paged services.

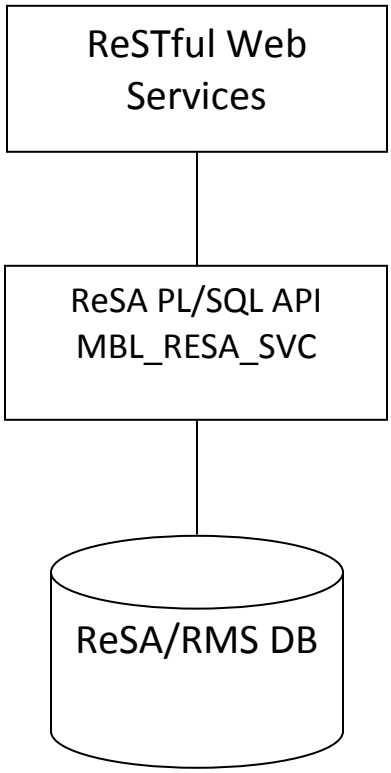
Each paged result will include the following information:

- **Total Record Count:** he number of all records matching the service input criteria.
- **Next Page URL**: he service URL with the same input parameters, but with the pageNumber plus 1 when more records exist.
- **Previous Page URL**: the service URL with same input parameters and the pageNumber input value minus 1 when the page number is not 1.

Both the next and pervious page URL will not be provided when:

- No records are returned.
- When the page number is 1 and there is no previous page.
- When last page of records and there is no next page.

Process Flow for the Web Service APIs



List of ReSTful Web Services

Following are the different ReSTful web services:

Summary of Open Store Days

Business Overview

This service provides, at a glance, the number of open stores for which the sales audit manager is responsible. The stores for which the user is responsible are those associated with the user in ReSA's employee maintenance through location traits.

Service Type

Get

ReST URL

`/summaryOpenStoreDay`

Input Parameters

No input.

Output

Record Type – DATE, OLDER, ALL

- For record type DATE: five records of type date are displayed for today minus 1 through today minus 5
- One record type OLDER: for store days older than today minus 5
- One record type ALL: for all store days

Record Date – Date of date type rows

Open Store Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_USER_LOC_TRAITS	Yes	No	No	No

Summary of Errors

Business Overview

This service provides, at a glance, the number outstanding errors on the specified days for stores for which the sales audit manager is responsible. An outstanding error is defined as an error that exists against a store day that has not been overridden.

Service Type

Get

ReST URL

`/summaryError`

Input Parameters

No input.

Output

Record Type – DATE, OLDER, ALL

- For record type DATE: five records of type date are displayed for today minus 1 through today minus 5
- One record type OLDER: for store days older than today minus 5
- One record type ALL: for all store days

Record Date – Date of date type rows

Error Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No

TABLE	SELECT	INSERT	UPDATE	DELETE
SA_USER_LOC_TRAITS	Yes	No	No	No

Summary of Over/Short Count

Business Overview

This service provides, at a glance, the count of overages and shortages for all open stores on a given day for which the sales audit manager is responsible. If the Over/Short value for the store day is a positive value, it is considered an overage. If the Over/Short value for the store day is a negative value, it is a shortage.

Service Type

Get

ReST URL

`/summaryOverShortCount`

Input Parameters

No input.

Output

Record Type – DATE, OLDER, ALL

- For record type DATE: five records of type date are displayed for today minus 1 through today minus 5
- One record type OLDER: for store days older than today minus 5
- One record type ALL: for all store days

Record Date – Date of date type rows

Over Count

Short Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
SA_HQ_VALUE	Yes	No	No	No
SA_POS_VALUE	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_SYS_VALUE	Yes	No	No	No
SA_TOTAL	Yes	No	No	No
SA_USER_LOC_TRAITS	Yes	No	No	No
STORE	Yes	No	No	No

Summary of Over/Short Amount

Business Overview

This service provides, at a glance, the sum of all overages and shortages for all open stores on a given day for which the sales audit manager is responsible. If all locations to which the user is responsible have the same local currency, all monetary values will be displayed in the local currency. Otherwise, all monetary values will be displayed in the retailer's primary currency. If the Over/Short value for the store day is a positive value, it is considered an overage. If the Over/Short value for the store day is a negative value, it is a shortage.

Service Type

Get

ReST URL

`/summaryOverShortAmount`

Input Parameters

No input.

Output

Record Type – DATE, OLDER, ALL

- For record type DATE: five records of type date are displayed for today minus 1 through today minus 5
- One record type OLDER: for store days older than today minus 5
- One record type ALL: for all store days

Record Date – Date of date type rows

Over Amount

Short Amount

Currency Code

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
MV_CURRENCY_CONVERSION_RATES	Yes	No	No	No
SA_HQ_VALUE	Yes	No	No	No
SA_POS_VALUE	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_SYS_VALUE	Yes	No	No	No
SA_TOTAL	Yes	No	No	No
SA_USER_LOC_TRAITS	Yes	No	No	No
STORE	Yes	No	No	No

Get Store Days

Business Overview

The service displays a list of open stores to which the user is assigned, for a single day, Older days, or All Days.

Service Type

Get

ReST URL

```
/getStoreDays?store={store}&recordType={recordType}&recordDate={recordDate}&sortAttrib={sortAttrib}&sortDirection={sortDirection}&pageSize={pageSize}&pageNumber={pageNumber}
```

Input Parameters

Parameter Name	Required	Description	Valid values
RecordType	Yes	Record Type	ALL, OLDER, DATE
RecordDate	No	Record Date, required when recordType is DATE	
Store	No	Store ID	
SortAttrib	No	Sort Attribute	STORENAME, AUDITOR, OSVALUE, ERRORCNT, DATASTATUS, OPENDAYS, OSDAYS and OSSUMS
SortDirection	No	Sort Direction	ASC, DESC
PageSize	No	Maximum number of locations to retrieve per page	
PageNumber	No	Result page to retrieve	

Output

Store

Store Day Seq No

Auditors

Business Date

Store Name

Chain

Chain Name

Data Status

Data Status Description

Audit Status

Audit Status Description
Audit Changed Datetime
Fuel Status
Fuel Status Description
Over Short Amount
Currency Code
Error Count
Transaction Count
Loaded File Count
Expected File Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_HQ_VALUE	Yes	No	No	No
SA_POS_VALUE	Yes	No	No	No
SA_STORE_DATA	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_SYS_VALUE	Yes	No	No	No
SA_SYSTEM_OPTIONS	Yes	No	No	No
SA_TOTAL	Yes	No	No	No
SA_TRAN_HEAD	Yes	No	No	No
SA_USER_LOC_TRAITS	Yes	No	No	No
V_CHAIN	Yes	No	No	No
V_CODE_DETAIL	Yes	No	No	No
V_STORE	Yes	No	No	No

Get Store Errors

Business Overview

Retrieves the summary of store day errors.

Service Type

Get

ReST URL

```
/getStoreErrors?store={store}&recordType={recordType}&recordDate={recordDate}
```

Input Parameters

Parameter Name	Required	Description	Valid values
RecordType	Yes	Record Type	ALL, OLDER, DATE
RecordDate	No	Record Date, required when recordType is DATE	
Store	No	Store ID	

Output

Store
Error Code
Error Description
Error Percentage

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SA_ERROR	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
V_SA_ERROR	Yes	No	No	No
V_STORE	Yes	No	No	No

Get Store Aggregations

Business Overview

Retrieves aggregated store day information for all dates or store days older than vdate -5.

Service Type

Get

ReST URL

```
/getStoreAggregations?allOlderInd={allOlderInd}&stores={stores}&sortAttrib={sortAttrib}&sortDirection={sortDirection}&pageSize={pageSize}&pageNumber={pageNumber}
```

Input Parameters

Parameter Name	Required	Description	Valid values
AllOlderInd	Yes	search string for locations Id or Name	ALL, OLDER
Stores	No	Comma Separated values for stores	

Parameter Name	Required	Description	Valid values
SortAttrib	No	Sort Attribute	STORENAME, AUDITOR, OSVALUE, ERRORCNT, DATASTATUS, OPENDAYS, OSDAYS and OSSUMS
SortDirection	No	Sort Direction	ASC, DESC
PageSize	No	Maximum number of locations to retrieve per page	
PageNumber	No	Result page to retrieve	

Output

Store
Store Name
Chain
Chain Name
Auditors
Open Days
Over Days
Short Days
Over Amount
Short Amount
Currency Code
Error Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SA_ERROR	Yes	No	No	No
SA_HQ_VALUE	Yes	No	No	No
SA_POS_VALUE	Yes	No	No	No
SA_STORE_DATA	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_SYS_VALUE	Yes	No	No	No
SA_TOTAL	Yes	No	No	No
V_CHAIN	Yes	No	No	No
V_STORE	Yes	No	No	No

Store Search

Business Overview

The web service enables store search and returns aggregated store information.

Service Type

Get

ReST URL

```
/storeSearch?searchString={searchString}&searchFilter={searchFilter}&sortAttrib={sortAttrib}&sortDirection={sortDirection}&pageSize={pageSize}&pageNumber={pageNumber}
```

Input Parameters

Parameter Name	Required	Description	Valid values
SearchString	Yes	Search string for locations ID or name	
SearchFilter	Yes	Search all stores or assigned stores	ALL, ASSIGN
SortAttrib	No	Sort attribute	STORENAME, AUDITOR, OSVALUE, ERRORCNT, DATASTATUS, OPENDAYS, OSDAYS and OSSUMS
SortDirection	No	Sort direction	ASC, DESC
PageSize	No	Maximum number of locations to retrieve per page	
PageNumber	No	Result page to retrieve	

Output

Store
Store Name
Chain
Chain Name
Auditors
Open Days
Over Days
Short Days
Over Amount
Short Amount
Currency Code
Error Count

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
LOC_TRAITS_MATRIX	Yes	No	No	No
SA_ERROR	Yes	No	No	No
SA_HQ_VALUE	Yes	No	No	No
SA_POS_VALUE	Yes	No	No	No
SA_STORE_DATA	Yes	No	No	No
SA_STORE_DAY	Yes	No	No	No
SA_SYS_VALUE	Yes	No	No	No
SA_TOTAL	Yes	No	No	No
SA_TRAN_HEAD	Yes	No	No	No
SA_USER_LOC_TRAITS	Yes	No	No	No
STORE	Yes	No	No	No
V_CHAIN	Yes	No	No	No
V_STORE	Yes	No	No	No

Get Store Day Date Indicator

Business Overview

This web service enables the user to find which store days have records that need attention.

Service Type

Get

ReST URL

```
/getStoreDateInd?store={store}
```

Input Parameters

Parameter Name	Required	Description
store	Yes	Store ID

Output

Record Type – DATE, OLDER, ALL

- For record type DATE: five records of type date are displayed for today minus 1 through today minus 5
- One record type OLDER: for store days older than today minus 5
- One record type ALL: for all store days

Record Date – Date of date type rows

Store Has Value Indicator

Table Impact

TABLE	SELECT	INSERT	UPDATE	DELETE
SA_STORE_DAY	Yes	No	No	No
V_STORE	Yes	No	No	No

8

Internationalization

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

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

Translation

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

- Graphical user interface (GUI)
- Error messages
- Reports

The following components are not translated:

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

The user interface for ReSA is available in these languages:

- Chinese (simplified)
- Chinese (traditional)
- Croatian
- Dutch
- French
- German
- Greek
- Hungarian
- Italian
- Japanese
- Korean
- Polish
- Portuguese (Brazilian)
- Russian
- Spanish

- Swedish
- Turkish

ReSA User Interface Language

When you first log into ReSA, the user interface will be set to the default language for that installation. The default language is specified in the <default-locale> tag in the faces-config.xml. You can change the language in the User Preferences screen.

There are two dropdown menus for language settings:

- Default
- Current Session

If you change the default language, that language will be the default language whenever you use ReSA. If you change the Current Session, that setting will only affect the current session.

The values for the language menus are set in the faces-config.xml file in the application archive file EarResaPortal.ear (located under the WEB-INF folder in the web archive WarResaPortal.war).

All user data, include default language, is stored internatlly in the ADF Metadata Stored (MDS) provided by the RAF platform library.

The User Interface strings are stored in a series of xlf files deployed with the application. The data translation for content fetched from the database like rtk_errors or code_detail uses the RMS translation tables like tl_shadow or code_detail_trans.

Unlike RMS which relies on the language code mapped to the database user id, ReSA uses the locale of the logged in application user. The language is set to the database session context RETAIL_CTX. APP_ISO_LANGUAGE. The base RMS data translation pacakges are changed to refer to RETAIL_CTX. APP_ISO_LANGUAGE if available, otherwise use database user's language to fetch the translated data.