Oracle® Retail Predictive Application Server

Release Notes

Release 13.0.3.8

April 2009

This document describes changes for Oracle Retail Predictive Application Server (RPAS) Release 13.0.3.8.

Documentation Release Only

> **Note:** The Oracle Retail Predictive Application Server (RPAS) 13.0.3.8 release is a documentation-only release. There is no code associated with this release identifier. RPAS 13.0.3 remains the current version of the code.

This Release Notes document is the only component of the RPAS 13.0.3.8 release. The purpose of this Release Notes document is only to document updates to the *Oracle Retail Predictive Application Server Administration Guide* for Release 13.0.3.

Retailers should refer to the following appendix in lieu of the corresponding sections in the "Appendix: Integration Guide" in the 13.0.3 *RPAS Administration Guide*.

Appendix: Integration Guide

RPAS and RDF Integration with RMS

Summary of Integration Approach with RMS

The strategy for the extraction of foundation data from RMS is for the extract programs (RMSE) to provide flat files in a generic format. For each solution that will use this data, transformation scripts are used to reformat the data as needed to produce a file suitable for loading into the application. For the instances of data coming from RPAS to non-RPAS applications, extract programs are specific to the application in need of the data. Other scripting languages are then used (Perl or AWK) to perform additional data formatting.

This appendix summarizes the following:

- RMS 13.0.2 to RDF 13.0.3 transformation programs
  - Merchandise Hierarchy
  - Location (Organization) Hierarchy
  - Calendar Hierarchy
  - Store Close Dates
  - Store Open Dates
  - Daily Sales and Issues
  - Weekly Sales and Issues
  - Out of Stock Indicator
- RDF 13.0.3 to RMS 13.0.2 extract programs
  - Approved Forecasts and Standard Deviations (Cumulative Intervals)
- Grade (RPAS 13.0.3) to RMS 13.0.2 extract programs
  - Store Grades
- Curve (RPAS 13.0.3) to Allocation 13.0.1 extract programs
  - Differentiator Profiles

Specifics on the usage of RMS extract programs (RMSE's) within the RDF transformation programs are beyond the scope of this document. See the *RMS Operations Guide* for more information on the RMS extract programs.

Environment Variable Setup

In addition to any variables identified in the RMS integration documentation, the transformation and/or extract programs require the following environment variables:

- RPAS_INTEGRATION_HOME: Identifies the location of the integration scripts when /common/header.ksh is run. This variable is used for all integration scripts packaged with the ARPOPlatform EXCEPT those included in 'rfx' (see $RDF_HOME below).
- $TO_RPAS: The staging area for the data to be loaded into RPAS. This directory should be located at the same level as the root of the RPAS domain. For example, if the domain RDF is located in Domains directory (example: /Domains/RDF), then $TO_RPAS should located at the same level as RDF (example: /Domains/to_rpas).
- $FROM_RPAS: The staging area for the data extract out of RPAS. This directory should be located at the same level as the root of the RPAS domain. For example, if the domain RDF is located in Domains directory (example: /Domains/RDF), then $FROM_RPAS should located at the same level as RDF (example: /Domains/from_rpas).
- $RDF_HOME: Identifies the location of the root of the 'rfx' directory. The 'rfx' directory packaged with the ARPOPlatform should be added to the location 'rfx' directory packaged with the RMS RETL programs.
- $RI_RMSVERSION: Identifies the version of RMS. If this variable is not set, the integration scripts assume an RMS version of 13. Set the value of this environment variable to 13.

RDF Transformation Programs

Common Program for All Transformations

The rdft.ksh script runs all of the necessary data extraction and transformation scripts (**rmse_*.ksh** and **rdft_*.ksh**, respectively) that are needed to produce the files to be loaded into RPAS/RDF/Planning. Most of these scripts are run in parallel (as background jobs).

Usage

rdft.ksh [-x] [-c] [-d dir]

Arguments:

- -x: This option will cause the execution of the RMS data extraction wrapper (rmse.ksh) to be skipped.
- -c: This option will cause FILE_DATE in rmse_config.env to be set to the current date instead of using VDATE.
- -d: This option will cause all programs executed by rdft.ksh to be obtained from the "dir" directory.

Transformations of Merchandise Hierarchy Data

rdft_merchhier.ksh is the primary script used to build the data for RPAS from the RMS Merchandise Hierarchy tables. The schema used to produce the output file depends on the attributes and differentiator settings in RMS:

- Case 1: If PROD_ATTRIBUTES_ACTIVE = False and DIFFS_ACTIVE = False, then rdft_merchhier.base.schema will be used to produce the file. In this case attributes and diff fields will not be included in the merchandise hierarchy file.
- Case 2: If PROD_ATTRIBUTES_ACTIVE = True and DIFFS_ACTIVE = False, then rdft_merchhier.attributes.schema will be used to produce the file. This schema must be manually edited to support a specific attribute model and must be kept in sync with rmse_attributes.schema and rmse_attributes.ksh (see the RMSE end user documentation).
- Case 3: If PROD_ATTRIBUTES_ACTIVE = False and DIFFS_ACTIVE = True, then rdft_merchhier.schema will be used to produce the file. In this case diff fields will be included in the merchandise hierarchy file.
- Case 4: If PROD_ATTRIBUTES_ACTIVE = True and DIFFS_ACTIVE = True, then an error will result. In this release, the combination of diffs and attributes is not supported.

Intermediate schema and scripts which may be used (depending on configuration options) to produce the merchandise hierarchy file:

- **rdft_diff.domain.schema**
- **rdft_merchdiff.domain.schema**
- **rdft_merchhier_diff_trans.ksh**
- **rdft_merchhier_split_by_domain.ksh**
- **rdft_clean_partition.ksh**

Additional merchandise hierarchy support for issue domains is provided in **rdft_item_loc.ksh**. This script is designed to produce a full item list for issues domains, only containing items that exist in the warehouses.

> **Note:** Issues-specific data transformation functionality is triggered based on the issues setting in RMS (ISSUES_ACTIVE must be set to True).

Transformations of Location Hierarchy Data

rdft_orghier.ksh is the primary script used to build the location data file needed for RPAS from the RMS Organizational Hierarchy Table.

The following five constants may be modified in the script based on location hierarchy data requirements:

- **COMPANY_NAME -** The label for the company position to be populated in the file.
- **COMPANY_ID** - The name for the company position to be populated in the file.
- **STORE_CLASS_CONCAT** - When set to "True", causes the STORE_CLASS to be concatenated on the left of the STORE_CLASS_DESCRIPTION field in the final Store data output file.
- **ADD_AT_SIGN_TO_WH_DESC** - When set to "True", will cause the WHSE_NAME field in the Warehouse ouput file to have an "@" prefix.
- **LONG_WAREHOUSE_RECORDS** - When set to "True", the Warehouse output records will consist of 16 fields. If it is "False", the records will contain only four fields, WH, WHSE_NAME, COMPANY and CO_NAME.

Intermediate schemas which may be used (depending on configuration options) to produce the location hierarchy file:

- rdft_issues.schema
- rdft_issues_long.schema
- rdft_orghier_store.schema

Note: Issues-specific data transformation functionality is triggered based on the issues setting in RMS (ISSUES_ACTIVE must be set to True).

Transformations of Calendar Hierarchy Data

rdft_calhier.ksh transforms the Calendar Hierarchy data extracted from RMS for loading into RPAS.

Configuration inputs to the script include:

- **DATE_PREF** - The path to the file that contains text indicating whether the format of the Date Description field will be mm/dd/yyyy or dd/mm/yyyy. See the RMS Operations Guide for date format options.
- **LAST_DOW** - The path to the file that contains a day of week name or abbreviation indicating which day of the week is considered to be the end of the week for the fiscal calendar being used at this installation.

Transformations of Daily Sales and Issues Data

rdft_daily_sales.ksh produces the daily sales and issues data files based on regular, promotion, clearance, and issues.

The following constant may be modified in the script based on data requirements:

- **DOM_START_COL** - Defines the starting column position of the Domain ID in the RETL output schema. This is needed by **rdft_merchhier_split_by_domain.ksh** to split the files by domain ID. If the OUTPUT_SCHEMA file is modified, the value of DOM_START_COL may also require modification from the default value.

Intermediate schemas which may be used (depending on configuration options) to produce the sales and/or issues data file:

- "rdft_daily_sales.schema

> **Note:** Issues-specific data transformation functionality is triggered based on the issues setting in RMS (ISSUES_ACTIVE must be set to True).

Transformations of Weekly Sales and Issues Data

rdft_weekly_sales.ksh produces the weekly sales and issues data files based on regular, promotion, clearance and issues.

The following constant may be modified in the script based on data requirements

- **DOM_START_COL** - Defines the starting column position of the Domain ID in the RETL output schema. This is needed by rdft_merchhier_split_by_domain.ksh to split the files by domain ID. If the OUTPUT_SCHEMA file is modified, the value of DOM_START_COL may also require modification from the default value.

Intermediate schemas which may be used (depending on configuration options) to produce the sales and/or issues data files:

- rdft_weekly_sales.schema

> **Note:** Issues-specific data transformation functionality is triggered based on the issues setting in RMS (ISSUES_ACTIVE must be set to True).

Transformations of Store Open Date Data

rdft_open_date.ksh produces the Store/Warehouse Opening Date data file.

Intermediate schema used to produce the store open date data files:

- rdft_open_date.schema

Transformations of Store Close Date Data

rdft_close_date.ksh produces the Store/Warehouse Closing Date data file.

Intermediate schema used to produce the store closing date data files:

- rdft_close_date.schema

Transformations of Out of Stock Indicator Data

rdft_outofstock.ksh produces the Store and Warehouse (issues) Out of Stock Indicator data extracted from RMS.

Intermediate schema and scripts which may be used (depending on configuration options) to produce the Out of Stock Indicator data file:

- rdft_outstock_split_by_domain.awk
- rdft_outofstock.schema
- rdft_outofstock_issues.schema
- rdft_outofstock_sales.schema

RDF Transformation Matrix

The following matrix identifies the transformation scripts and schemas used for each the hierarchy and data files produced for RDF 13.0.2.

Table 1 RDF Transformation Matrix

Directory	Script or Schema Name	Merchandise Hierarchy	Location (Organization) Hierarchy	Calendar	Daily Sales & Issues	Weekly Sales & Issues	Out of Stock Indicator	Store Open Dates	Store Close Dates
rfx/lib	rdft_merchhier_diff_trans.ksh	X							
	rdft_merchhier_split_by_domain.ksh	X							
	rdft_outofstock_split_by_domain.ksh						X		

Table 1 (Cont.) RDF Transformation Matrix

Directory	Script or Schema Name	Merchandise Hierarchy	Location (Organization) Hierarchy	Calendar	Daily Sales & Issues	Weekly Sales & Issues	Out of Stock Indicator	Store Open Dates	Store Close Dates
rfx/schema	rdft_close_date.schema								X
	rdft_daily_sales.schema				X				
	rdft_diff.domain.schema	X							
	rdft_merchierdiff.domain.schema	X							
	rdft_merchier.attributes.schema	X							
	rdft_merchhier.base.schema	X							
	rdft_merchhier.domain.schema	X							
	rdft_merchhier.schema	X							
	rdft_open_date.schema							X	
	rdft_orghier_issues.schema		X						
	rdft_orghier_issues_long.schema		X						
	rdft_orghier_store.schema		X						
	rdft_outofstock.schema						X		
	rdft_outofstock_issues.schema						X		
	rdft_outofstock_sales.schema						X		
	rdft_weekly_sales.schema					X			

Table 1 (Cont.) RDF Transformation Matrix

Directory	Script or Schema Name	Merchandise Hierarchy	Location (Organization) Hierarchy	Calendar	Daily Sales & Issues	Weekly Sales & Issues	Out of Stock Indicator	Store Open Dates	Store Close Dates
rfx/src	rdft_ksh	X	X	X	X	X	X	X	X
	rdft_calhier.ksh			X					
	rdft_clean_partition.ksh	X							
	rdft_close_date.ksh								X
	rdft_daily_sales.ksh				X				
	rdft_item_loc.ksh	X							
	rdft_merchhier.ksh	X							
	rdft_open_date.ksh							X	
	rdft_orghier.ksh		X						
	rdft_outofstock.ksh						X		
	rdft_weekly_sales.ksh					X			

Common Programs for Extracts

config.ksh is a configuration directory that requires both the RMS version being integrated and the backup action to be defined.

The following OPTIONAL arguments are available:

- Name of the domain: Defaults to directory name
- Number of the domain: Defaults to the 2 last digits of the directory name
- Format of timestamp attached to logs and processed input files: Defaults to: (date +"%b%d%a%I%M%p") (example: Aug02Thu0111PM)
- Data Drop: Defaults to ../../to_rpas
- Data Export: Defaults to ../../from_rpas
- Log Drop: Defaults to ./logs
- Error Drop: Defaults to ./err
- Reclass Data: Defaults to ../reclass_data

functions.ksh

This script file contains ksh functions that are used by scripts in [DOM]/scripts. It should be sourced, not executed in order to preserve environment variables.

header.ksh

This script should be run at the beginning of any implementation-specific script to setup function libraries, environment, and platform-specific routines.

Extract of Forecast Data for RMS

rdf_e_rms.ksh extracts forecast demand value and standard deviation (cumulative interval) at both day and week aggregations from an RDF domain.

Arguments:

- -t: <Domain Type> (S for sales, I for issues)
- -w: <Data Width> ([7...18], defaults to 12)
- -d: <Domain> (defaults to current directory)
- -n: <Domain Number> (defaults to last two digits of domain)

Output files:

${RPAS_EXPORT}/d<s|i>demand.<Domain Number> (demand at day)

${RPAS_EXPORT}/w<s|i>demand.<Domain Number> (demand at week)

The following table provides information about the output file data format.

Table 2 Output File Data Format

Field	Start	Width	Format
Day | EOW Day	1	8	Alpha
Product ID	1	25	Alpha
Location ID	26	20	Alpha
Demand	46	12	Alpha
Std. Dev. Demand	68*	12*	Numeric (floating point, 4 decimal digits with decimal)

* Width of Demand and Std. Dev. Demand may be overridden with the -w parameter; stated values Demand width and Std. Dev. Demand start and width are based on default width of 12.

> **Note:** the following must be defined in the shell environment prior to calling this script:
>
> - RPAS_HOME
>
> - RPAS_INTEGRATION_HOME

Load of Extracted Forecast Data and Standard Deviations to RMS

rmsl_forecast.ksh pulls the daily/weekly forecast items into RMS.

During the loading of each domain file the following steps are performed:

1. Truncate the partition in the RMS forecast table which corresponds to the domain ID.

   > **Note:** Partition names should always be in the format: [tablename]_[domainID]

2. Append a domain field and insert the domain_id into each record.
3. Load the forecast data into the RMS forecast table.

   Example: `rmsl_rpas_forecast.ksh  daily | weekly`

Intermediate schemas which may be used (depending on configuration options) to produce the forecast data files:

- rmsl_forecast_daily.schema
- rmsl_forecast_weekly.schema

Extract of Diff Profile Data for Allocation

profile_e_alloc.ksh extracts Curve diff profiles for use by Allocation (an RMS application).

Arguments:

- -p: <Profile Number>
- -m: <Mask Measure> (Optional mask; only positions for which the mask value is non-NA will be exported.)
- -w: <Data Width> ([7...18], defaults to 12)
- -d: <Domain> (defaults to current directory)
- -n: <Domain Number> (defaults to last two digits of domain)

Output file: ${RPAS_EXPORT}/d1<Product Level>.<Domain Number>, where Product Level is the Aggregation intersection's Prod dimension.

The following table provides information about the output file data format.

Table 3 Output File Data Format

Field	Start	Width	Format
Product ID	1	25	Alpha
Location ID	26	20	Alpha
Diff ID (optional)	46	36	Alpha
Quantity	82	12*	Numeric (floating point, 4 decimal digits, no decimal)*
Std. Dev. Demand	68*	12*	Numeric (floating point, 4 decimal digits with decimal)

* Quantity width may be overridden with the -w parameter.

Note: The following must be defined in the shell environment prior to calling this script:

- RPAS_HOME

- RPAS_INTEGRATION_HOME

Extract of Store Grade Data for RMS

grade_e_rms.ksh extracts store grades for use by RMS.

Arguments:

- -t <Timestamp> (YYMMDDTTTT). This value corresponds to the timestamp of the Cluster Membership measure (clpm+<Timestamp>) to be extracted
- -d <Domain> (defaults to current directory)
- -n <Domain Number> (defaults to last two digits of domain)

Output file: ${RPAS_EXPORT}/gr<Timestamp>.<Domain Number>

Output file data format:

- (a thru c) constitutes the header records
- (d thru j) constitutes the detail records
- (k thu l) constitutes the footer records

  a. ’FHEAD'

  b. Line ID Number

  c. ’GRADU'

  d. ’FDETL' (Record Identifier)

  e. Line Sequence Identifier

  f. Grade Group ID Number: This value corresponds to the first 8 characters of the Cluster Run Name measure (clnam+<user-defined name>) set by the user in the Generate Cluster wizard in Grade. For integration with RMS, the Cluster Run Name must be populated with only numeric characters.

  g. Grade Group: This value corresponds to the first "N" characters of the Cluster Run Name measure (clnam+<user-defined name>) set by the user in the Generate Cluster wizard in Grade. "N" is 20 for RMS version 11.0, and "N" is 120 for RMS version 12 and beyond. The script determines the RMS version from the environment variable RI_RMSVERSION. RMS version 12 is assigned by default.

  h. Store ID

  i. Grade Member Name

  j. 'FTAIL' (Record Identifier)

  k. Line ID Number

  l. FDETL Line Total Number

Note: The following must be defined in the shell environment prior to calling this script:

- RPAS_HOME

- RPAS_INTEGRATION_HOME

RDF Extract Matrix

The following matrix identifies the extract scripts and schemas used for each the data files produced for RMS:

Table 4 RDF Extract Matrix

Directory	Script or Schema Name	Forecasts and Standard Deviations	Diff Profiles
common	config.ksh		
	functions.ksh	X	
	header.ksh	X	X
curve	profile_e_alloc.ksh		X
grade	grade_e_rms.ksh		
plan	Plan_e_alloc.ksh		
	Plan_e_price.ksh		
	Plan_e_plcblwdm.ksh		
	Plan_e_ploblwdm.ksh		
rdf	rdf_e_rms.ksh	X	
	rmsl_forecast.ksh	X	
	rmsl_forecast_ daily.schema	X	
	rmsl_forecast_ weekly.schema	X	

Integration with Oracle Retail Workspace

The Oracle Retail Workspace installer prompts you to enter the URL for your supported Oracle Retail applications. However, if a client installs a new application after Oracle Retail Workspace is installed, the retail-workspace-page-config.xml file needs to be edited to reflect the new application.

The file as supplied comes with all appropriate products configured, but the configurations of non-installed products have been "turned off". Therefore, when "turning on" a product, locate the appropriate entry, set "rendered" to "true", and enter the correct URL and parameters for the new application.

The entry consists of the main URL string plus one parameter named "config". The value of the config parameter is inserted by the installer. Somewhere in the installer property files there is a value for the properties "deploy.retail.product.rms.url" and "deploy.retail.product.rms.config".

The entry consists of the main URL string plus one parameter named "config". The value of the config parameter will be inserted by the installer. Somewhere in the installer property files there will be a value for the properties "deploy.retail.product.rms.url" and "deploy.retail.product.rms.config".

For example, suppose RMS was installed on mycomputer.mycompany.com, port 7777, using a standard install and rms configured with the application name of "rms13sedevhpsso". If you were to access RMS directly from your browser, you would type in:

```
http://mycomputer.mycompany.com:7777/forms/frmservlet?config=rms
13sedevhpsso
```

The entry in the retail-workspace-page-config.xml after installation would resemble the following:

```
<url>http://mycomputer.mycompany.com:7777/forms/frmservlet</url>

  <parameters>
     <parameter name="config">
        <value>rms13sedevhpsso</value>
     </parameter>
  </parameters>
```

Oracle Single Sign-on Overview

Single Sign-On (SSO) is a term for the ability to sign onto multiple Web applications via a single user ID/Password. There are many implementations of SSO - Oracle currently provides three different implementations: Oracle Single Sign-On (OSSO), Java SSO (with the 10.1.3.1 release of OC4J) and Oracle Access Manager (provides more comprehensive user access capabilities).

Most, if not all, SSO technologies use a session cookie to hold encrypted data passed to each application. The SSO infrastructure has the responsibility to validate these cookies and, possibly, update this information. The user is directed to log on only if the cookie is not present or has become invalid. These session cookies are restricted to a single browser session and are never written to a file.

Another facet of SSO is how these technologies redirect a user's Web browser to various servlets. The SSO implementation determines when and where these redirects occur and what the final screen shown to the user is.

Most SSO implementations are performed in an application's infrastructure and not in the application logic itself. Applications that leverage infrastructure managed authentication (such as deploying specifying "Basic" or "Form" authentication) typically have little or no code changes when adapted to work in an SSO environment.

What Do I Need for Oracle Single Sign-On?

The nexus of an Oracle Single Sign-On system is the Oracle Identity Management Infrastructure installation. This consists of the following components:

- An Oracle Internet Directory (OID) LDAP server, used to store user, role, security, and other information. OID uses an Oracle database as the back-end storage of this information.
- An Oracle Single Sign-On servlet, used to authenticate the user and create the OSSO session cookie. This servlet is deployed within the infrastructure Oracle Application Server (OAS).
- The Delegated Administration Services (DAS) application, used to administer users and group information. This information may also be loaded or modified via standard LDAP Data Interchange Format (LDIF) scripts.
- Additional administrative scripts for configuring the OSSO system and registering HTTP servers.

Additional OAS servers will be needed to deploy the business applications leveraging the OSSO technology.

Can Oracle Single Sign-On Work with Other SSO Implementations?

Yes, OSSO has the ability to interoperate with many other SSO implementations, but some restrictions exist.

Authentication

Authentication is the process of establishing a user's identity. There are many types of authentication. The most common authentication process involves a user ID and password.

Dynamically Protected URLs

A "Dynamically Protected URL" is a URL whose implementing application is aware of the OSSO environment. The application may allow a user limited access when the user has not been authenticated. Applications that implement dynamic OSSO protection typically display a "Login" link to provide user authentication and gain greater access to the application's resources.

Identity Management Infrastructure

The Identity Management Infrastructure is the collection of product and services which provide Oracle Single Sign-on functionality. This includes the Oracle Internet Directory, an Oracle HTTP server, and the Oracle Single Sign-On services. The Oracle Application Server deployed with these components is typically referred as the "Infrastructure" instance.

MOD_OSSO

mod_osso is an Apache Web Server module an Oracle HTTP Server uses to function as a partner application within an Oracle Single Sign-On environment. The Oracle HTTP Server is based on the Apache HTTP Server.

Oracle Internet Directory

Oracle Internet Directory (OID) is an LDAP-compliant directory service. It contains user ids, passwords, group membership, privileges, and other attributes for users who are authenticated using Oracle Single Sign-On.

Partner Application

A partner application is an application that delegates authentication to the Oracle Identity Management Infrastructure. One such partner application is the Oracle HTTP Server (OHS) supplied with the Oracle Application Server. OHS uses the MOD_OSSO module to configure this functionality.

All partner applications must be registered with the Oracle Single Sign-On server. An output product of this registration is a configuration file the partner application uses to verify a user has been previously authenticated.

Realm

A Realm is a collection users and groups (roles) managed by a single password policy. This policy controls what may be used for authentication (for example, passwords, X.509 certificates, and biometric devices). A Realm also contains an authorization policy used for controlling access to applications or resources used by one or more applications.

A single OID can contain multiple Realms. This feature can consolidate security for retailers with multiple banners or to consolidate security for multiple development and test environments.

Statically Protected URLs

A URL is considered to be "Statically Protected" when an Oracle HTTP server is configured to limit access to this URL to only SSO authenticated users. Any attempt to access a "Statically Protected URL" results in the display of a login page or an error page to the user.

Servlets, static HTML pages, and JSP pages may be statically protected.

What Single Sign-On is Not

Single Sign-On is NOT a user ID/password mapping technology.

However, some applications can store and retrieve user IDs and passwords for non-SSO applications within an OID LDAP server. An example of this is the Oracle Forms Web Application framework, which maps OSSO user IDs to a database logins on a per-application basis.

How Oracle Single Sign-On Works

Oracle Single Sign-On involves a couple of different components. These are:

- The Oracle Single Sign-On (OSSO) servlet, which is responsible for the back-end authentication of the user.
- The Oracle Internet Directory LDAP server, which stores user IDs, passwords, and group (role) membership.
- The Oracle HTTP Server associated with the Web application, which verifies and controls browser redirection to the OSSO servlet.
- If the Web application implements dynamic protection, then the Web application itself is involved with the OSSO system.

Statically Protected URLs

When an unauthenticated user accesses a statically protected URL, the following occurs:

1. The Oracle HTTP server recognizes the user has not been authenticated and redirects the browser to the Oracle Single Sign-On servlet.
2. The OSSO servlet determines the user must authenticate, and displays the OSSO login page.
3. The user must sign in via a valid user ID and password. If the OSSO servlet has been configured to support multiple Realms, a valid realm must also be entered. The user ID, password, and realm information is validated against the Oracle Internet Directory LDAP server.
4. The OSSO servlet creates and sends the user's browser an OSSO session cookie. This cookie is never persisted to disk and is specific only to the current browser session. This cookie contains the user's authenticated identity. It does NOT contain the user's password.
5. The OSSO servlet redirects the user back to the Oracle HTTP Server, along with OSSO specific information.
6. The Oracle HTTP Server decodes the OSSO information, stores it with the user's session, and allows the user access to the original URL.

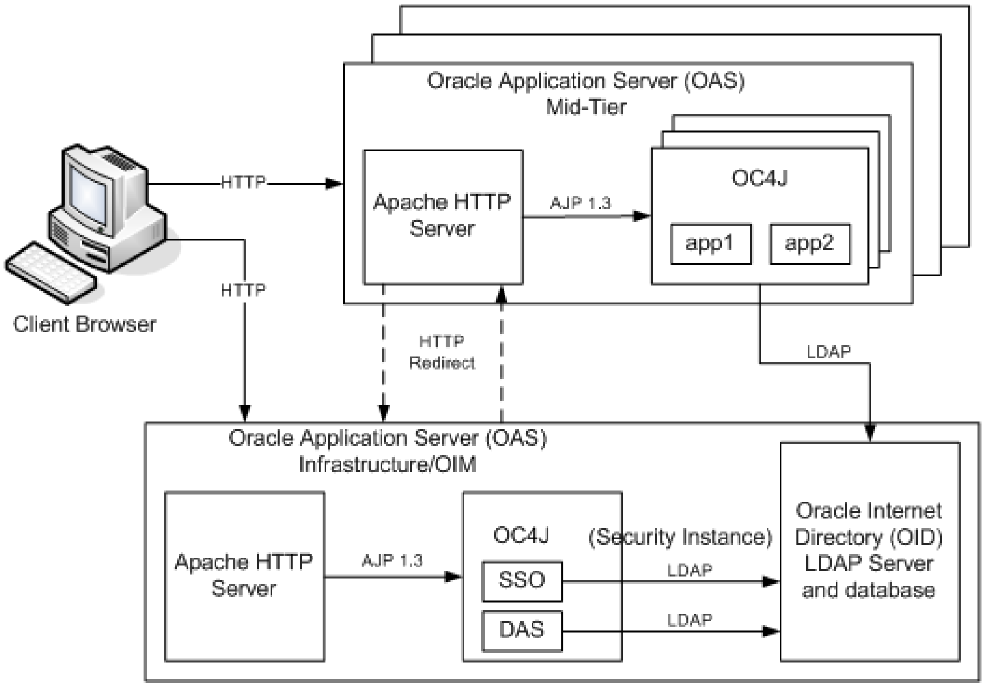
Dynamically Protected URLs

When an unauthenticated user accesses a dynamically protected URL, the following occurs:

1. The Oracle HTTP server recognizes the user has not been authenticated, but allows the user to access the URL.
2. The application determines the user must be authenticated and sends the Oracle HTTP server a specific status to begin the authentication process.
3. The Oracle HTTP Server redirects the user's browser session to the OSSO Servlet.
4. The OSSO servlet determines the user must authenticate, and displays the OSSO login page.
5. The user must sign in via a valid user ID and password. If the OSSO servlet has been configured to support multiple Realms, a valid realm must also be entered. The user ID, password, and realm information is validated against the Oracle Internet Directory LDAP server.
6. The OSSO servlet creates and sends the user's browser an OSSO session cookie. This cookie is never persisted to disk and is specific only to the current browser session. This cookie contains the user's authenticated identity. It does NOT contain the user's password.
7. The OSSO servlet redirects the user back to the Oracle HTTP Server, along with OSSO specific information.
8. The Oracle HTTP Server decodes the OSSO information, stores it with the user's session, and allows the user access to the original URL.

Single Sign-on Topology

Figure 1 Single Sign-on Topology



Installation Overview

Installing Oracle Single Sign-On consists of installing the following components:

1. Installing the Oracle Internet Directory (OID) LDAP server and the Infrastructure Oracle Application Server (OAS). These are typically performed using a single session of the Oracle Universal Installer and are performed at the same time. OID requires an Oracle relational database and if one is not available, the installer will also install this as well.

   The Infrastructure OAS includes the Delegated Administration Services (DAS) application as well as the OSSO servlet. The DAS application can be used for user and realm management within OID.

2. Installing additional midtier instances (such as OAS 10.1.2) for the Oracle Retail applications, such as RMS, that are based on Oracle Forms technologies. These instances must be registered with the Infrastructure OAS installed in step 1).

3. Installing additional application servers to deploy other Oracle Retail applications and performing application specific initialization and deployment activities.

Infrastructure Installation and Configuration

The Infrastructure installation for OSSO is dependent on the environment and requirements for its use. Deploying an Infrastructure OAS to be used in a test environment does not have the same availability requirements as for a production environment. Similarly, the Oracle Internet Directory (OID) LDAP server can be deployed in a variety of different configurations. See the *Oracle Application Server Installation Guide* and the *Oracle Internet Directory Installation Guide* for more details.

OID User Data

Oracle Internet Directory is an LDAP v3 compliant directory server. It provides standards-based user definitions out of the box.

The current version of Oracle Single Sign-On only supports OID as its user storage facility. Customers with existing corporate LDAP implementations may need to synchronize user information between their existing LDAP directory servers and OID. OID supports standard LDIF file formats and provides a JNDI compliant set of Java classes as well. Moreover, OID provides additional synchronization and replication facilities to integrate with other corporate LDAP implementations.

Each user ID stored in OID has a specific record containing user specific information. For role-based access, groups of users can be defined and managed within OID. Applications can thus grant access based on group (role) membership saving administration time and providing a more secure implementation.

OID with Multiple Realms

OID and OSSO can be configured to support multiple user Realms. Each realm is independent from each other and contains its own set of user IDs. As such, creating a new realm is an alternative to installing multiple OID and Infrastructure instances. Hence, a single Infrastructure OAS can be used to support many development and test environments by defining one realm for each environment.

Realms may also be used to support multiple groups of external users, such as those from partner companies. For more information on Realms, see the *Oracle Internet Directory Administrators Guide.*

User Management

User Management consists of displaying, creating, updating or removing user information. There are two basic methods of performing user management: LDIF scripts and the Delegate Administration Services (DAS) application.

OID DAS

The DAS application is a Web-based application designed for both administrators and users. A user may update their password, change their telephone number of record, or modify other user information. Users may search for other users based on partial strings of the user's name or ID. An administrator may create new users, unlock passwords, or delete users.

The DAS application is fully customizable. Administrators may define what user attributes are required, optional or even prompted for when a new user is created.

Furthermore, the DAS application is secure. Administrators may also what user attributes are displayed to other users. Administration is based on permission grants, so different users may have different capabilities for user management based on their roles within their organization.

LDIF Scripts

Script based user management can be used to synchronize data between multiple LDAP servers. The standard format for these scripts is the LDAP Data Interchange Format (LDIF). OID supports LDIF script for importing and exporting user information. LDIF scripts may also be used for bulk user load operations.

User Data Synchronization

The user store for Oracle Single Sign-On resides within the Oracle Internet Directory (OID) LDAP server. Oracle Retail applications may require additional information attached to a user name for application-specific purposes and may be stored in an application-specific database. Currently, there are no Oracle Retail tools for synchronizing changes in OID stored information with application-specific user stores. Implementers should plan appropriate time and resources for this process. Oracle Retail strongly suggests that you configure any Oracle Retail application using an LDAP for its user store to point to the same OID server used with Oracle Single Sign-On.

Oracle® Retail Predictive Application Server Release Notes, Release 13.0.3.8