Oracle® Retail Extract, Transform, and Load

Installation Guide

Release 13.2.4

E28078-01

March 2012

ORACLE®

Oracle ® Retail Extract, Transform, and Load Installation Guide, Release 13.2.4

E28078-01

Primary Author: Susan McKibbon

Contributing Author: Gloreen Soans

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

Retail Extract, Transform, and Load Installation Guide, Release 13.2.4

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

The Retail Extract, Transform, and Load (RETL) Installation Guide contains a complete description of the RETL installation process and configuration instructions.

Audience

This guide is intended for system administrators and assumes that you are familiar with the following:

- Installing, configuring, and managing the application server software and security.
- Installing, configuring, and managing the relational database management systems. You must be familiar with the Database Administrator (DBA) level commands and tasks.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html if you are hearing impaired.

Related Documentation

For more information, see the following document in the Retail Extract, Transform, and Load Release 13.2.4 documentation set:

- *Oracle Retail Extract, Transform, and Load Release Notes*
- *Oracle Retail Extract, Transform, and Load Programmer's Guide*

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, 13.2) or a later patch release (for example, 13.2.4). If you are installing the base release, additional patch, and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

Note: For releases earlier than Release 13.2.4, the installation instructions are available in the *Oracle Retail Extract, Transform, and Load Programmer's Guide*.

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/technology/documentation/oracle_retail.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

The following text conventions are used in this document:

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

1

Overview

This chapter provides an overview of Retail Extract Transform and Load (RETL) and a roadmap to install RETL.

Overview

RETL is a high-performance, scalable, platform-independent, parallel processing data movement tool. RETL addresses several primary needs:

- Database-independent applications
- Platform-independent applications
- Developing applications more quickly than possible with conventional coding methods (such as custom-crafted C or C++ code)
- High-performance data processing

To provide for these needs, RETL defines an interface in XML that applications can call to define Extraction, Transformation, and Loading (ETL) functions. This interface is in a well-defined XML form that allows access to any database that RETL supports.

RETL is a cross-platform development tool supported on Oracle Linux. The XML definitions do not change on a per-platform basis, so moving between hardware platforms is simplified.

Development of the XML instructions for RETL is simpler, faster, and less error-prone than writing C or C++ code. Applications can be completed much faster than possible with previous methods.

RETL Installation Roadmap

The installation roadmap lists the high-level steps for installing and configuring RETL. These instructions include documentation links that you can access to get more details about each task.

To install RETL, perform the following steps:

1. Prepare your system environment for the installation. Ensure that you have access to the supported database server and software certifications required for RETL to work properly. For more information, see Chapter 2, "Preinstallation Tasks".
2. Set up password stores. For more information, see Chapter 3, "Installing RETL".
3. Review the installation order. For more information, see "Appendix: Installation Order."
4. When your system is ready for the installation, you can begin installing RETL.

   Ensure that you have access to the installation software. For more information on installing RETL, see Chapter 3, "Installing RETL".
5. Verify that the installation is complete. For more information on validating your installation, see the Validating the RETL Instllation section in Chapter 3, "Installing RETL".
6. After the installation is complete and validated, you can upgrade to the latest available version of RETL. For more information on upgrading RETL, see Upgrading from an Earlier Version of RETL section in Chapter 3, "Installing RETL".
7. You can now customise the configuration file as per your requirements. For more information on configuring RETL, see Chapter 4, "Configuring RETL".

2

Preinstallation Tasks

This chapter describes the technical specifications necessary for installing RETL.

Checking the Technical Specifications

RETL is certified on the platforms listed in this section.

Note: For historical reasons, the executable for RETL is called rfx.

Supporting the RAC Database

RETL supports only THIN database driver connection to connect to Real Application Cluster (RAC) and non RAC databases. It does not support Oracle Call Interface (OCI) Database driver connection.

RETL database operators are modified to support RAC databases. (For non RAC databases, no modifications are required to RETL database operators and RETL scripts.)

jdbcconnectionstring

For the RAC database, you must add the jdbcconnectionstring property to all database operators (ORAREAD, ORAWRITE, UPDATE, DELETE, INSERT, and PREPAREDSTATEMENT). The value for this property is RAC URL.

The following is an example of the value used for the jdbcconnectionstring property.

```
<PROPERTY name="jdbcconnectionstring" value="jdbc:oracle:thin:@(DESCRIPTION
=(ADDRESS_LIST =(ADDRESS = (PROTOCOL = TCP)(HOST = hostname1)(PORT =
1521))(ADDRESS = (PROTOCOL = TCP)(HOST = hostname2)(PORT = 1521))(LOAD_BALANCE =
yes ))(CONNECT_DATA =(SERVICE_NAME = serviceName)))"/>
```

Note: For answers to frequently asked questions about the jdbcconnectionstring property and for RAC database examples, see the *Oracle Retail Extract, Transform, and Load Programmer's Guide*.

Supported Database Server Requirements

Supported On	Versions Supported
Database Server Operating System	Operating System certified with Oracle Database 11gR2 Enterprise Edition: ■ Oracle Linux 5 Update 5 for x86-64 (actual hardware or Oracle virtual computer) ■ Red Hat Enterprise Linux 5 Update 5 (RHEL 5.5) for x86-64 (actual hardware or Oracle virtual computer) ■ IBM AIX 6.1 (actual hardware or LPARs) ■ Solaris 10 Sparc (actual hardware or logical domains) ■ HP-UX 11.31 Integrity (actual hardware, HPVM, or vPars)
Database Server 11gR2	Oracle Database Enterprise Edition 11gR2 (11.2.0.2) with the following specifications: **Components**: ■ Oracle Partitioning ■ Examples CD (Formerly the companion CD) **One-off Patches**: ■ 10170431—CTWR consumes a lot of CPU cycles. If you use ASM, apply the following patch to database home. ■ 11808931—Merge request on top of 11.2.0.2.0 for defects 10410054 and 10422126 **Other components**: ■ Perl compiler 5.0 or later ■ X-Windows interface

Supported Oracle Retail Products

RETL is a standalone product and does not necessarily require any products installed for it to function properly. The following Oracle Retail products can be integrated with RETL.

Product	Version
Oracle Retail Merchandising System (RMS)	13.2.4
Oracle Retail Advanced Inventory Planning (AIP)	13.2.4
Oracle Retail Predictive Application Server (RPAS)	13.3

UNIX User Account Privileges to Install the Software

A UNIX user account is needed to install the RETL software. The UNIX user account that you use for installing the software must have write access for the computer.

Storing Credentials

Database user names and passwords are stored in an encrypted format inside a wallet file (`cwallet.sso`) using the CSF API under `$RFX_HOME/etc/security` directory structure. You must not delete the wallet file, as it contains credentials required for RETL functionality.

To set up database credentials during installation, use the following utility: `setup-security-credential.sh`. This utility is in the `$RFX_HOME/bin` directory.

To support backward compatibility, the following utility is available: `credential-convert-utility.sh`. This utility is in the `$RFX_HOME/bin` directory and is used to convert all pre-13.2 RETL flow scripts to RETL 13.2.4 compatible flow scripts.

3

Installing RETL

After you have prepared your system for installation, you can install RETL as described in this chapter.

Overview of the Installation Process

Install RETL on each server system involved in the input, output, or processing of data. For example, if System A outputs data files and System B inputs and processes that data, install RETL on both System A and System B.

Accessing the Installation Package

You can access the installation package from the My Oracle Support Web site (https://support.oracle.com).

Installing RETL

To install RETL, perform the following steps:

1. Log in as the `root` user on the host.
2. Create a UNIX group for the rfx - group that owns the RETL software.
3. Create a UNIX operating system account on the appropriate host, using ksh as the default shell.

   ```
   rfx - rfx group
   ```

4. Create a directory where you will install this software.
5. Log in to the UNIX server as `rfx`.
6. Download `rfx-13.2.4.tar` (the installation package) and place the RETL installation package on the UNIX server under the directory, `rfxdownload1324`, which was created in Step 4.
7. Extract `rfx-13.2.4.tar` under the same directory:

   ```
   >cd rfxdownload1324
   >tar -xvf rfx-13.2.4.tar
   ```

8. Change directories to the location where the package is installed.

   ```
   >cd rfxdownload1324/rfx-13.2.4
   ```

9. At the UNIX prompt, enter:

```
>./install.sh
```

10. Follow the prompts to install RETL for your configuration.

```
$ ./install.sh
Enter directory for RETL software:
```

> **Note:** This directory must differ from the one created in Step 4.

```
---> <enter path to RFX_HOME>
Is this the correct directory for the install? y or n
RFX_HOME: <path to RFX HOME>
---> y
Creating RFX_HOME directory <path to RFX_HOME>...
Creating install directory in <path to RFX_HOME> ...
Copying Library Files...
Copying Sample Files...
Copying Executables...
Copying Config File...
Successful completion of RETL Install

To complete the RETL setup and installation:
1) Place the following in a .kshrc/ .profile to retain setup variables:
         RFX_HOME=<RFX HOME path>
         JAVA_HOME=<JAVA HOME path>
         PATH=$RFX HOME/bin:$JAVA_HOME/bin:$PATH
2) Be sure to verify any additional environment setup as per the "Setup"
section of the Programmers Guide.
3) Verify the installation by running the following command:

$RFX_HOME/bin/retl -version
retl 13.2.4 build 2063
```

11. Install Java Runtime Environment 1.6 (JRE).

    JRE 1.6 is required for RETL to function properly on Sun Solaris, IBM AIX, HP-UX, and Oracle Linux platforms. If it is not included with your operating system installation, you must install it separately. You can download the Java Runtime Environment 1.6 (JRE) for the corresponding platforms at the Web sites listed below:

    For Sun Solaris: http://java.sun.com/javase/downloads/index.jsp

    For IBM AIX: http://www.ibm.com/developerworks/java/jdk/aix/service.html

    For HP-UX: http://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HPUXJAVAHOME

    For Oracle Linux: http://java.sun.com/javase/index.jsp

12. Review the `install.log` file in the *<base directory>*`/install` directory to verify that RETL was installed successfully.

Note: Users cannot specify database credentials (such as user name and password) in flow scripts for operators such as ORAREAD, DBREAD, DB2READ, TERAFEREAD, ORAWRITE, DB2WRITE, TERAWRITE, UPDATE, DELETE, INSERT, DBLOOKUP, and PREPAREDSTATEMENT.

13. To store the credentials (username/password) for a particular database, execute `setup-security-credential.sh` in the `$RFX_HOME/bin` directory for RETL installation.

    This utility provides options to Add, Modify, and Delete the database credentials. While adding a database credential, it prompts for "dbuseralias," which must be unique for each database credential because it is as a primary key.

    Each unique dbuseralias prompts for the username and password. Once provided, this information is stored in the wallet file (`cwallet.sso`) under `$RFX_HOME/etc/security` directory for RETL installation.

14. Provide dbuseralias value for the dbuseralias tag in the flowscript for the particular database information. For example, <PROPERTY name="dbuseralias" value="rms1323dbuseralias"/>

    Note: Ensure that the wallet file (`cwallet.sso`) is not deleted, as it stores the required credentials for proper RETL operation.

15. Use the Modify and Delete utility functions to modify or delete the credentials for a particular database.

16. For Unix OS, set up the environment variables. For more information, see "Post Installation Setup".

17. For Linux OS, create symbolic link for gsort and set up environment variables. For more information, see "Creating Symbolic Link for gsort on Linux OS".

18. Verify the installation and setup by running the "verify_retl" script.

Creating Symbolic Link for gsort on Linux OS

On the Linux platforms, a link can be created in the `$RFX_HOME/bin` to link the default Linux sort to gsort. This is done as follows.

- Use the "which sort" command to find where sort resides on the Linux box.
- Create a symbolic link for gsort in the `$RFX_HOME/bin` folder pointing to `$RFX_HOME/bin/gsort`.

For example, assuming that on the Linux box the sort resides under `/usr/bin/sort`, create the symbolic link for gsort pointing to `$RFX_HOME/bin/gsort` as in -s `/usr/bin/sort $RFX_HOME/bin/gsort` under the `$RFX_HOME/bin` folder.

Set up an Environment Variable for Linux OS

You must set an environment variable to ensure that RETL runs properly on Linux OS. The following example shows the variable you must set up in your Linux profile.

```
export LC_ALL=C
```

Post Installation Setup

To set up your UNIX environment, perform the following:

1. After installing RETL, setup your UNIX environment for RETL. The following example shows the variables you need in your UNIX profile to run RETL properly.

   ```
   export RFX_HOME=<base directory>
   export JAVA_HOME=<enter java 1.6 home here>
   export PATH=$RFX_HOME/lib:$RFX_HOME/bin:$JAVA_HOME/bin:${PATH}
   ```

2. If you have selected an installation with database support, set up database and environment variables required for basic database setup. Refer to the *Oracle Retail Extract, Transform, and Load Programmer's Guide* for information on setting up your database.

   ```
   export ORACLE_HOME=/your/Oracle_home/directory
   export PATH=$ORACLE_HOME/bin;$PATH
   ```

3. Log in to the UNIX server as rfx. At the UNIX prompt, enter the following:

   ```
   >rfx
   ```

4. Make changes as necessary to the operator defaults in `rfx.conf`. For example, Oracle database operators require hostname/port to be specified. These changes can be made to `rfx.conf` as DEFAULTS for convenience.
5. Make changes as necessary to the temporary directory settings in `rfx.conf`.
6. Make changes as necessary to the default performance log file location configured in `logger.conf`.

   The distributed `logger.conf` specifies `/tmp/rfx.log` as the default performance log file. If this file cannot be created, the log4j logging facility reports an error.

   To change the performance log file location, modify `logger.conf` and change the value of the name parameter in the PERFORMANCE-APPENDER element. The line has the following format:

   ```
   <param name="file" value="/tmp/rfx.log"/>
   ```

   For example, to change the performance log file location to /var/tmp/rfx/rfx-performance.log, change the line as follows:

   ```
   <param name="file" value="/var/tmp/rfx/rfx-performance.log"/>
   ```

7. If RETL is installed correctly and the .profile is correct, the following results:

   ```
   Error : Flow file argument ('-f') required!
   ```

Upgrading from an Earlier Version of RETL

RETL releases are required to be backward compatible with the XML flow interface of earlier releases. To upgrade RETL, perform the following steps:

1. Choose a new (and different) location to install the new version of RETL. For more information on installing RETL, see "Installing RETL".
2. Change the environment in which RETL runs so that it refers to the new RFX_HOME (for example, change within .kshrc or .cshrc).
3. Double check the environment variables to make sure that you were not explicitly referring to old RETL directories.
4. Update flows so that they are compatible for the current release. For more information, refer th*e Oracle Retail Extract, Transform, and Load Programmer's Guide*.

   Note: Users cannot specify database credentials (such as user name and password) in flow scripts for operators such as ORAREAD, DBREAD, DB2READ, TERAFEREAD, ORAWRITE, DB2WRITE, TERAWRITE, UPDATE, DELETE, INSERT, DBLOOKUP, and PREPAREDSTATEMENT.

5. To store the credentials (username/password) for a particular database, execute `setup-security-credential.sh` in the $RFX_HOME/bin directory for RETL installation.

   This utility provides options to Add, Modify, and Delete the database credentials. While adding a database credential, it prompts for "dbuseralias," which must be unique for each database credential because it is similar to a primary key.

   The following is a usage example for the `credential-convert-utility.sh` utility.

   ```
   >./credential-convert-utility.sh -inDir <Path for input files> -outDir <Path
   for output files>

   Where <Path for input files> - Path to Input directory containing the pre
   13.2.4 Release flow scripts.
                  <Path for output files> - Path to Output directory for
   RETL-13.2.4 compatible flow scripts to be generated.
   ```

   Note: The directory path for input files must differ from the directory path for output files.

6. Verify that your flows and scripts run as expected. If any scripts or flows fail to run with the new release, but pass in running with the old release, and changes are not otherwise noted in the *Oracle Retail Extract, Transform, and Load Release Notes*, contact Customer Support. For more information, see "Customer Support"

To make future upgrades easier, separate the RETL-specific changes that you make to your environment, so that the environment variables can be easily removed, modified, or replaced when necessary.

Validating the RETL Installation

When setting up RETL in a new environment (or if you would just like to verify that the RETL environment is set up properly) run the verify_retl script located in the /bin directory of the RETL installation. (The verify_retl script is available as of release 10.2.)

> **Note:** RFX_HOME should be set properly prior to running `verify_retl`.

The RETL package verification tool performs the following checks:

- Verifies environment variables/etc is set up properly.
- Ensures that the RETL binary is installed properly and can run.
- Runs a series of system tests derived from the samples directory.
- Logs information about environment setup in a log file.

The usage for verify_retl is as follows:

```
verify_retl [-doracle] [-nodb] [-h]
```

Option	Description
-doracle	Checks environment variables for the Oracle installation of RETL.
-nodb	Checks environment variables for the stand-alone version of RETL.
-h	Displays the help message.

This generates the following output if successful:

```
Checking RETL Environment...found ORACLE environment...passed!
Checking RETL binary...passed!
Running samples...passed!
========================================================================
Congratulations! Your RETL environment and installation passed all tests. See the
programmer's guide for more information about how to further test your database
installation (if applicable)
========================================================================
Exiting...saving output in /files0/retl/tmp/verifyretl-20384.log
```

Check the RETL ENVIRONMENT SETUP section of the log file for important information on environment variables that must be set.

Understanding the RETL Backward Compatibility Options

A major requirement for RETL releases is that they be backward-compatible in the XML interface with earlier releases. There are a few small changes that must be made for certain operators when upgrading to RETL 11.x, RETL 12.x, or RETL 13.x from RETL 10.x releases.

XML Flow Interface/Operator Differences Between 10.x and Later Releases

Operator	Property	Backward Compatibility Notes
FILTER	filter	DEPRECATED SYNTAX—RETL 10.x versions produced warning messages to correct 'filter' syntax. Later versions do not accept the following syntax in the filter property : >, <, >=, <=, =. These operations are replaced by GT, LT, GE, LE, EQ, respectively.

ORAREAD	query	INVALID SYNTAX—RETL 10.x versions allowed input of invalid XML in the query property of the dbread operators. Characters such as '>' and '<' in the query property will cause later versions to produce an error message.
		An example follows:
		Previous XML valid in 10.x versions:
		`<PROPERTY name="query" value="SELECT * FROM ANY_ TABLE WHERE ANY_COLUMN > 1" />`
		Property should now appear as the following in later versions:
		`<PROPERTY name="query">`
		`<![CDATA[`
		`SELECT * FROM ANY_TABLE WHERE ANY_COLUMN > 1`
		`]]>`
		`</PROPERTY>`

ORAREAD/ ORAWRITE	dbname	The database to which to connect. This property should be used instead of sid from now on.
	port	The port on which the database listener resides. **Note:** For Oracle databases, use tnsping to obtain the port number. The default for Oracle is 1521.This may need to be specified only once in the `rfx.conf` configuration file for convenience.
ORAWRITE	hostname	The fully specified hostname or IP address where the database resides. **Note:** This property should only be specified when it is known that connections are being made to a remote database. This may need to be specified only once in the `rfx.conf` configuration file for convenience.

Operator	Property	Backward Compatibility Notes
■ ORAREAD ■ DBREAD ■ DB2READ ■ TERAFEREAD ■ ORAWRITE ■ DB2WRITE ■ TERAWRITE ■ UPDATE ■ DELETE ■ INSERT ■ DBLOOKUP ■ PREPARED STATE MENT	connectstring	These properties are no longer valid.
■ ORAREAD ■ DBREAD ■ DB2READ ■ TERAFEREAD ■ ORAWRITE ■ DB2WRITE ■ TERAWRITE ■ UPDATE ■ DELETE ■ INSERT ■ DBLOOKUP ■ PREPARED STATE MENT	dbuseralias	Used to store the credentials (user name/password) for a particular database.

Examples (in operator for ORAWRITE):

Connection to database through THIN database driver:

```
<OPERATOR type="orawrite">
    <OUTPUT name="test.v"/>
         <PROPERTY name="dbname"          value="databasename"/>
         <PROPERTY name="dbuseralias" value=" dbuserAliasName"/>
         <PROPERTY name="tablename"       value="mytable"/>
         <PROPERTY name="createtablemode" value="recreate"/>
         <PROPERTY name="hostname"        value="myhostname"/>
         <PROPERTY name="port"            value="1521"/>
</OPERATOR>
```

Identifying Hardware Requirement Differences Between RETL 10 and Later Releases

In general, more physical memory is required in order to run the later versions of RETL than in RETL 10. There is no general formula or guideline for the additional memory requirement, because it strongly correlates to the flow, data, configuration, and so on.

rfx Command Line Options

You can get help on rfx options on the command line by entering the following on the command line:

```
rfx -h
```

The output is like the following:

```
>rfx -h
rfx [ OPTIONS ]

-h          Print help and exit
-oOPNAME    Print operator help. Valid values:
              operator name or 'ALL' for all ops
-e          Print RETL environment variable usage
-v          Print version and exit
-cFILE      Configuration File
-nNUMPARTS  Number of Partitions (SMP only)
-x          Disable partitioning (default=off)
-sTYPE      Display schema as TYPE. Valid values:
              NONE,SCHEMAFILE (default=NONE)
-lLOGFILE   Log statistics/times to LOGFILE
-fFLOWFILE  XML file containing flow
-d          Produce daVinci files (default=off)
-g          Produce flow graphs (default=off)
```

These options are described in the following table:

Option	Default Value	Description
-h		Shows the help message shown above.
-oOPNAME		Displays syntax usage for operator specified in OPNAME, or for all operators if OPNAME is "ALL". Valid operator names are the same as those operators used in the XML flow interfaces. The intention with this option is to provide online syntax help for flow developers, reducing the need to refer to this document for syntax usage. See the *Oracle Retail Extract, Transform, and Load Programmer's Guide* for more information about this option.
-e		Prints RETL environment variables that can be used for things such as turning on verbose debugging, setting JVM parameters, and so on.
-v		Displays the version and build number.
-cSTRING	$RFX_HOME/etc/rfx.conf	Overrides the default configuration file.
-nINT	As specified in the rfx.conf , or 1 if no rfx.conf is found	The number of partitions to use. This feature is intended for RETL experts only.
-x	Partitioning as defined in the rfx.conf	Disables partitioning.
-sTYPE	NONE	Prints the input and output schemas for each operator. Valid values and descriptions: NONE—rfx will not print any schema information. SCHEMAFILE—If specified, this option prints the input and output for each operator in schema file format so that developers can quickly and easily cut and paste rfx output to a file and break up flows. Developers could then modify these files for the purposes of specifying IMPORT and EXPORT schema files. **Note:** rfx should be run with the `-sNONE` option in production systems where unnecessary output is not needed. The -sSCHEMAFILE option is often useful in development environments where it is desirable to debug RETL flows by breaking them up into smaller portions.
-1LOGFILE	N/A	Specifies the log file in which to log RETL statistics/times. If the log file path is relative, the log file is placed in the directory as defined in the TEMPDIR element of the RETL configuration file (`rfx.conf`). This changes the default log file as specified in `rfx.conf` and will turn on logging only if the log level in `rfx.conf` is set to 1 or more.

Option	Default Value	Description

-fSTRING	N/A	Specifies the file to use as input to rfx. This is where the XML flow is located. If no file is specified, rfx will read from standard input. The following is the syntax to use when reading from stdin via the korn shell (ksh):
		`rfx -f - <<EOF`
		`<FLOW>`
		…
		`</FLOW>`
		`EOF`

--davinci-files	Off	This option is not currently supported.
-g	Off	Produce visual graphs of a flow. **Note:** The flow is not run. See the *Oracle Retail Extract, Transform, and Load Programmer's Guide* for more information on how to use this option.

For more information on DOTTY, see the following:
http://www.research.att.com/sw/tools/graphviz/download.html

RETL Environment Variables

You can retrieve a list of environment variables with the following ksh syntax:

```
export VARIABLE=VALUE
```

Option	Description
RFX_DEBUG	Option to turn on verbose RETL debugging to standard output. Set to 1 to turn on. Default is 0.
RFX_SHOW_SQL	Option to turn on verbose RETL database debugging to standard output. Set to 1 to turn on. Default is 0.
RETL_ENABLE_ ASSERTIONS	Option to enable assertion checking in RETL (use only when there appears to be a bug in RETL itself). Set to 1 to turn on. Default is 0.
RETL_VM_MODE	highvol, lowvol Volume option. Set to highvol to turn on JVM options for RETL in high volume environments with longer-running processes. Default is highvol and this should not be changed unless the flow has been shown to run faster in "lowvol" mode.
RETL_INIT_HEAP_SIZE	xxxM, where xxx is a number in megabytes Setting for the initial heap size for the Java Virtual Machine (JVM). Default is 50M.
RETL_MAX_HEAP_SIZE	xxxM, where xxx is a number in megabytes Setting for the maximum heap size for the Java Virtual Machine (JVM). Default is 300M.
RETL_JAVA_HOME	Any path to a valid Java Runtime Environment (JRE) Option to reset the location of the Java Runtime Environment (JRE).

Option	Description
JAVA_ARGS	Valid JVM arguments Option to set any JVM parameters. These are placed on the command-line as arguments to the 'java' command. This option should not be used unless instructed to do so by Oracle Retail Support, or if the user is aware of the implications of setting JVM parameters and has tested the results of making any changes.
RETL_TMP	Set for RETL flow temporary files (for example, temporary flow files from "here" documents). Defaults to "." or the local directory where the script is run from. This can be either an absolute path name or relative to your current directory.
RETL_ENABLE_64BIT_ JVM	Option to enable 64bit JVM. Set to 1 to enable a 64bit JVM if available for the particular platform.

4

Configuring RETL

After installing RETL, you must follow the configuration steps listed in this chapter.

Configuration

You can specify a configuration file to control how RETL uses system resources, where to store temporary files, and to set default values for operators.

Configuration Field Descriptions

Use the following descriptions to modify your configuration file located in the `<base_directory>/etc` directory.

Element Name	Attribute Name	Attribute Value/Description
CONFIG		CONFIG is the root element of the RETL Configuration file. This element can have either NODE or DEFAULTS elements.
NODE		NODE is a child element of the CONFIG element.
	hostname	The name of the UNIX server where RETL is installed.
	bufsize	The number of records allowed between operators at any given time. The default bufsize is 2048 records. The bufsize can have a significant impact on performance. Setting this value too low causes a significant amount of contention between processes, slowing down RETL. A value that is too high can also slow RETL down because it will consume more memory than it needs to. Finding the right value for your hardware configuration and flow is part of tuning with RETL.
	numpartitions	Optional value, defaults to 1. See the following sections on partitioning.
TEMPDIR		TEMPDIR is a child element of the NODE element. This element can be specified more than one time. RETL will use the temporary file directories in a round-robin fashion.
	path	Path to the directory where the RETL writes temporary files.

Element Name	Attribute Name	Attribute Value/Description
GLOBAL		The GLOBAL element is a child of CONFIG and specifies values for global settings within RETL.
	bytes_per_character	This setting specifies the maximum number of bytes that are in a character. Setting this value is necessary for allowing RETL to work with UNICODE data.
DEFAULTS		This element is a child of the CONFIGURATION element. This section is used to define one or more default PROPERTY values for operators that are reused frequently. Care should be taken when using and changing these defaults since they can change the results of a RETL flow without changing individual flows.
	operator	Operator type Name of the operator to assign default values. Refer to the following chapters for the operators that are available.
PROPERTY		The PROPERTY element is a child of the DEFAULTS element.
	name	The name of the operator property to assign a default value. See the following chapters for the property names for each operator.
	value	The value assigned as the default for the specified operator property. See the following chapters for the property values for each operator.

Element Name	Attribute Name	Attribute Value/Description
LOGGER		The LOGGER element specifies a facility for RETL performance logging. This gives a dynamic view of RETL to allow developers to get some information about the data that flows through each operator. This also enables developers to determine if/when deadlock conditions occur, debug problems and tune performance. See the properties that follow on how to configure the LOGGER for RETL.
	type	"file" The type of logging facility to use. Currently the only value RETL allows is to log to a file.
	dest	An optional output destination to log to. Currently, this value must be an absolute or relative filename. If the filename is relative, the log file will be placed in the directory as defined in the TEMPDIR element of the RETL configuration file (`rfx.conf`). The log file can be overridden by specifying -lLOGFILE as a command-line parameter to rfx. The default value is "`rfx.log`". **Note:** The dest filename overrides the first performance log file location specified in `logger.conf`.
	level	0, 1, 2 Specifies the level of detailed information recorded in the log file. Higher levels mean more information is logged. Log level values have the following meaning: "0" = no logging. "1" = logging of operator start times. "2" = logging of the above plus: ■ Operator end times ■ Record count per operator ■ Records per second per operator ■ Start and stop times for major events that are performed by RETL (for example, query execution, database utility execution, and external sorts). If the flow name is provided in the FLOW element, it is logged with each logger entry When the log level is set to "2", a performance report in HTML format will also be written at the end of each RETL run. This report will show hotspots in the RETL flow, in actual time spent per operator. **Note:** Leave logging turned off in production systems where performance is a critical factor. The logging feature is not designed to log errors, but is intended to give rough measures of flow performance characteristics and aid in debugging flows. It is recommended to periodically remove the log file to free disk space.

The following is a sample resource configuration for RETL:

```
<CONFIG>
    <NODE hostname="localhost" numpartitions="1" bufsize="2048" >
         <TEMPDIR path="/u00/rfx/tmp"/>
         <TEMPDIR path="/u01/rfx/tmp"/>
    </NODE>
    <GLOBAL bytes_per_character="1" >
         <LOGGER type="file" dest="rfx.log" level="0" />
    </GLOBAL>
    <DEFAULTS operator="oraread">
         <PROPERTY name="maxdescriptors" value="100"/>
         <PROPERTY name="hostname" value="mspdev25"/>
         <PROPERTY name="port" value="1521"/>
    </DEFAULTS>
    <DEFAULTS operator="orawrite">
         <PROPERTY name="hostname" value="mspdev25"/>
         <PROPERTY name="port" value="1521"/>
    </DEFAULTS>
</CONFIG>
```

Temporary Space Configuration

The best performance can probably be attained when the number of temporary directories specified in the configuration file is equal to the number of partitions. Ideally, each temp directory should be on a separate disk controller.

> **Note:** These directories must always be local to the host where RETL is running. Use of network drives can have a drastic impact on performance.

By default, TEMPDIR is set to /tmp. This should be changed to be a local disk after installation, because /tmp on many platforms is a memory device used for operating system swap space. If this is not changed, the system could be exhausted of physical memory.

Take care to protect the files within this directory, because they can contain userid and password information. Check with your system administrator about setting the permissions so that only the appropriate personnel can access these files (for example, by setting: umask 077).

Temporary files should be removed from temporary directories on a daily or weekly basis. This important server maintenance task aids in RETL debugging, reviewing database-loading utility log files, and other activities. If a RETL module fails, you should not rerun the module until after removing the temporary files that were generated by the failing module.

Logger Configuration

Logging in to RETL is performed using the log4j logging facility. log4j is a flexible open-source package maintained by the Apache Software Foundation.

log4j can be configured to send logged output to the console, a file, or even a file that automatically rolls over at a given frequency. log4j can also send logged information to more than one destination. Additionally, log4j loggers have a tunable logging level that can control how much information is logged. For more information about log4j, refer to the following:

http://logging.apache.org/log4j/docs/documentation.html

RETL uses two log4j loggers. The first logger is the performance logger. The second logger, the output logger, handles RETL output that is normally sent to the terminal.

Performance Logger

The performance logger logs operator statistics such as start time, stop time, and records processed per second. It also records the start and stop time of various system events, such as sorts and database queries.

The performance logger is configured in the `logger.conf` file located in the <base_ directory>etc directory. The performance logger's log4j name is `retek.retl.performance`.

To turn on performance logging, edit `logger.conf` and find the logger XML element where the name attribute is retek.retl.performance. Change the level to DEBUG. (By default, the level is WARN.)

Performance information is logged to `/tmp/rfx.log`. To change the location of this file, change the file specified in the PERFORMANCE-APPENDER appender.

Note: If a file is specified in the LOGGER element in the rfx.conf file, it overrides the first file specified in the `logger.conf` file.

Output Logger

The output logger logs informational, warning, and error messages. By default, all of these messages are written to the terminal. You can configure the output logger to change the destination for these messages.

If you want to use any of the advanced features of log4j, change the output logger settings in `logger.conf`. The log4j name of the output logger is retek.retl.

For example, if you want to log all errors into a file and also display them on the terminal, make the following changes to the `logger.conf` file:

1. Add the following before the logger element for the retek.retl logger, replacing file-name-for-RETL-errors with the name of the file you want to use:

   This step creates an appender. Appenders tell the log4j logging system where and how to log messages. The above example tells log4j that the ERRORSTOFILE appender is a file appender that writes to the file specified in the file parameter. The pattern layout shows how to format the message, and the filter dictates that only messages logged at level ERROR or above are logged.

   ```
   <appender name="ERRORSTOFILE" class="org.apache.log4j.FileAppender">
   <param name="file" value="file-name-for-RETL-errors"/>
   <layout class="org.apache.log4j.PatternLayout">
   <param name="ConversionPattern" value="[%d] %-5p - %m%n"/>
   </layout>
   <filter class="org.apache.log4j.varia.LevelRangeFilter">
   <param name="acceptOnMatch" value="true"/>
   <param name="levelMin" value="ERROR"/>
   <param name="levelMax" value="FATAL"/>
   </filter>
   </appender>
   ```

2. Add the following to the `retek.retl` logger element to associate the ERRORSTOFILE appender with the `retek.retl` logger.:

   ```
   <appender-ref ref="ERRORSTOFILE"/>
   ```

Multibyte Character Support

The bytes_per_character setting in the configuration file allows RETL to provide multibyte character support. While this variable is optional, it is required to ensure proper processing of data containing multibyte characters.

The setting of bytes_per_character is used mainly in the parsing of the schemas used by IMPORT and EXPORT. The schema files allow field length specification in the number of characters; RETL uses the bytes_per_character setting to convert the field length from a number of characters into a number of bytes.

The field length is used in several other operators, so the setting of bytes_per_character indirectly affects those operators as well. Most notably, it affects ORAWRITE. When ORAWRITE is configured to create or recreate the target table, RETL uses the field lengths to define the length of the table columns. For example, if bytes_per_character is set to 3, and a field is specified with a length of 10 characters in the IMPORT schema, the resulting database column will be 30 bytes wide.

No corresponding conversion takes place when data is read from an Oracle database, because the database reports the field lengths in bytes.

Note: The Oracle database reports the length of string constants in a query in the number of characters, instead of the number of bytes as it does with the table columns. RETL does not correctly compensate for the different units of measurement. Using a string constant with multibyte characters in a query is therefore highly discouraged.

For example, the example query below contains a string constant with multibyte characters in FIELD1. The Oracle database reports the length of the FIELD1 field as 2 characters instead of 6 bytes, so RETL has an incorrect field length for FIELD1.

```
SELECT
    '店舗' FIELD1,
    FIELD2
FROM
      TABLE1
```

A

Appendix: Setting Up Password Stores with Oracle Wallet

As part of an application installation, administrators must setup password stores for database user accounts using Oracle Wallet. You must install these password stores on the application database side. While the installer handles much of this process, the administrators must perform some additional steps.

You must also install a password store for the application and application server user accounts; however, the installer takes care of this entire process.

Password Stores and Oracle Wallet

Oracle databases allow other users on the server to see passwords in case database connect strings (username/password@db) are passed to programs. In the past, users could navigate to `ps -ef|grep <username>` and see the password if the password was supplied in the command line when calling a program.

To make passwords more secure, Oracle Retail has implemented the Oracle Software Security Assurance (OSSA) program. Sensitive information such as user credentials now must be encrypted and stored in a secure location. This location is called password stores or wallets. These password stores are secure software containers that store the encrypted user credentials.

Users can retrieve the credentials using aliases that were set up when encrypting and storing the user credentials in the password store. For example, if username/password@db is entered in the command line argument and the alias is called db_username, then the argument to a program would be the following:

```
sqlplus /@db_username
```

This would connect to the database as it did previously, but it would hide the password from any system user.

After this is configured, as in the example above, the application installation and the other relevant scripts no longer need to use embedded usernames and passwords. This reduces any security risks that may exist because usernames and passwords are no longer exposed.

When the installation starts, all the necessary user credentials are retrieved from the Oracle Wallet based on the alias name associated with the user credentials.

There are two different types of password stores or wallets. One type is for database connect strings used in program arguments (such as sqlplus /@db_username). The other type is for java application installation and application use.

Setting Up Password Stores for Database User Accounts

After the database is installed and the default database user accounts are set up, administrators must set up a password store using the Oracle Wallet. This involves assigning an alias for the username and associated password for each database user account. The alias is used later during the application installation. This password store must be created on the system where the application server and database client are installed.

This section describes the steps you must take to set up a wallet and the aliases for the database user accounts. For more information on configuring authentication and password stores, refer to the *Oracle Database Security Guide*.

Note: In this section, `<wallet_location>` is a placeholder text for illustration purposes. Before running the command, ensure that you specify the path to the location where you want to create and store the wallet.

To set up a password store for the database user accounts, perform the following steps:

1. Create a wallet using the following command:

   ```
   mkstore -wrl <wallet_location> -create
   ```

   After you run the command, a prompt appears. Enter a password for the Oracle Wallet in the prompt.

   Note: The `mkstore` utility is included in the Oracle Database Client installation.

   The wallet is created with the auto-login feature enabled. This feature enables the database client to access the wallet contents without using the password. For more information, refer to the *Oracle Database Advanced Security Administrator's Guide.*

2. Create the database connection credentials in the wallet using the following command:

   ```
   mkstore -wrl <wallet_location> -createCredential <alias-name>
   <database-user-name>
   ```

   After you run the command, a prompt appears. Enter the password associated with the database user account in the prompt.

3. Repeat Step 2 for all the database user accounts.

4. Update the `sqlnet.ora` file to include the following statements:

   ```
   WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY =
   <wallet_location>)))
   SQLNET.WALLET_OVERRIDE = TRUE
   SSL_CLIENT_AUTHENTICATION = FALSE
   ```

5. Update the `tnsnames.ora` file to include the following entry for each alias name to be set up.

```
<alias-name> =
    (DESCRIPTION =
     (ADDRESS_LIST =
            (ADDRESS = (PROTOCOL = TCP) (HOST = <host>) (PORT = <port>))
        )
        (CONNECT_DATA =
            (SERVICE_NAME = <service>)
         )
     )
```

In the previous example, `<alias-name>`, `<host>`, `<port>`, and `<service>` are placeholder text for illustration purposes. Ensure that you replace these with the relevant values.

Setting Up Wallets for Database User Accounts

The following examples show how to set up wallets for database user accounts for the following applications:

- For RMS, RPM Plsql Batch, RETL DB, RWMS batch, and ARI
- For Java Applications (SIM, ReIM, RPM, Alloc, RIB, RSL, AIP, RETL)

For RMS, RPM Plsql Batch, RETL DB, RWMS batch, and ARI

1. Create a new directory called wallet under your folder structure.

```
cd /projects/rms13.2/dev/
mkdir .wallet
```

> **Note:** The default permissions of the wallet allow only the owner to use it, ensuring the connection information is protected. If you want other users to be able to use the connection, you must adjust permissions appropriately to ensure only authorized users have access to the wallet.

2. Create a `sqlnet.ora` in the wallet directory with the following content.

```
WALLET_LOCATION =   (SOURCE =     (METHOD = FILE)     (METHOD_DATA =
(DIRECTORY =  /projects/rms13.2/dev/.wallet)) )
SQLNET.WALLET_OVERRIDE=TRUE
SSL_CLIENT_AUTHENTICATION=FALSE
```

> **Note:** WALLET_LOCATION must be on line 1 in the file.

3. Set up a `tnsnames.ora` in the wallet directory. This `tnsnames.ora` includes the standard `tnsnames.ora` file. Add two custom tns_alias entries that are only for use with the wallet. For example, sqlplus /@dvols29_rms01user.

```
ifile = /u00/oracle/product/11.2.0.1/network/admin/tnsnames.ora

dvols29_rms01user =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = mspdv311.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SID = dvols29) (GLOBAL_NAME = dvols29)))

dvols29_rms01user.world =
  (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = tcp)
  (host = mspdv311.us.oracle.com) (Port = 1521)))
    (CONNECT_DATA = (SID = dvols29) (GLOBAL_NAME = dvols29)))
```

> **Note:** It is important to not just copy the `tnsnames.ora` file because it can quickly become out of date. The ifile clause (shown above) is key.

4. Create the wallet files. These are empty initially.
   a. Ensure you are in the intended location.

```
$ pwd
/projects/rms13.2/dev/.wallet
```

   b. Create the wallet files.

```
$ mkstore -wrl . -create
```

   c. Enter the wallet password you want to use. It is recommended that you use the same password as the UNIX user you are creating the wallet on.
   d. Enter password again.

   Two wallet files are created from the above command: `ewallet.p12` and `cwallet.sso`.

5. Create the wallet entry that associates the username and password to the custom tns alias that was setup in the wallet's `tnsnames.ora` file.

```
mkstore -wrl . -createCredential <tns_alias> <username> <password>
```

> **Example:** mkstore -wrl . -createCredential dvols29_rms01user rms01user passwd

6. Test the connectivity. The ORACLE_HOME used with the wallet must be the same version or higher then what the wallet was created with.

```
$ export TNS_ADMIN=/projects/rms13.2/dev/.wallet /* This is very import to use
wallet to point at the alternate tnsnames.ora created in this example */

$ sqlplus /@dvols29_rms01user

SQL*Plus: Release 11
```

```
Connected to:
Oracle Database 11g

SQL> show user
USER is "rms01user"
```

Running batch programs or shell scripts would be similar:

```
Ex: dtesys /@dvols29_rms01user
script.sh /@dvols29_rms01user

Set the UP unix variable to help with some compiles :

export UP=/@dvols29_rms01user
for use in RMS batch compiles, and RMS, RWMS, and ARI forms compiles.
```

As shown in the example above, users can ensure that passwords remain invisible.

Additional Database Wallet Commands

The following is a list of additional database wallet commands.

- Delete a credential on wallet

  ```
  mkstore -wrl . -deleteCredential dvols29_rms01user
  ```

- Change the password for a credential on wallet

  ```
  mkstore -wrl . -modifyCredential dvols29_rms01user rms01user passwd
  ```

- List the wallet credential entries

  ```
  mkstore -wrl . -list
  ```

  This command returns values such as the following:

  ```
  oracle.security.client.connect_string1
  oracle.security.client.user1
  oracle.security.client.password1
  ```

- View the details of a wallet entry

  ```
  mkstore -wrl . -viewEntry oracle.security.client.connect_string1
  ```

  Returns the value of the entry:

  ```
  dvols29_rms01user
  mkstore -wrl . -viewEntry oracle.security.client.user1
  ```

  Returns value of the entry:

  ```
  rms01user

  mkstore -wrl . -viewEntry oracle.security.client.password1
  ```

  Returns value of the entry:

  ```
  passwd
  ```

For Java Applications (SIM, ReIM, RPM, Alloc, RIB, RSL, AIP, RETL)

For Java applications, consider the following.

- For database user accounts, ensure that you set up the same alias names between the password stores (database wallet and Java wallet). You can provide the alias name during the installer process.
- Document all aliases that you have set up. During the application installation, you must enter the alias names for the application installer to connect to the database and application server.
- Passwords are not used to update entries in Java wallets. Entries in Java wallets are stored in partitions, or application-level keys. In each retail application that has been installed, the wallet is located in `<WEBLOGIC_DOMAIN_HOME>/retail/<appname>/config`.

  For example:

  ```
  mspdv351:[1033_WLS] /u00/webadmin/product/10.3.x/WLS/user_projects/
  domains/132_mck_soa_domain/retail/reim13/config
  ```

- Application installers should create the Java wallets for you, but it is good to know how this works for future use and understanding.
- Scripts are located in `<WEBLOGIC_DOMAIN_HOME>/retail/<appname>/retail-public-security-api/bin` for administering wallet entries.

  For example:

  ```
  mspdv351:[103x_WLS] /u00/webadmin/product/10.3.3/WLS/user_projects/
  domains/132_mck_soa_domain/retail/reim13/retail-public-security-api/bin
  ```

- In this directory is a script to help you update each alias entry without having to remember the wallet details. For example, if you set the RPM database alias to rms01user, you will find a script called `update-RMS01USER.sh`.

  Note: These scripts are available only with applications installed by way of an installer.

- Two main scripts are related to this script in the folder for more generic wallet operations: `dump_credentials.sh` and `save_credential.sh`.
- If you have not installed the application yet, you can unzip the application zip file and view these scripts in `<app>/application/retail-public-security-api.bin`.

  For example:

  ```
  mspdv351:[103x_WLS]
  /u00/webadmin/reim/application/retail-public-security-api/bin
  ```

update-<ALIAS>.sh

update-<ALIAS>.sh updates the wallet entry for this alias. You can use this script to change the user name and password for this alias. Because the application refers only to the alias, no changes are needed in application properties files.

Usage:

```
update-<username>.sh <myuser>
```

Example:

```
mspdev71:[103x4WLS] /u00/webadmin/product/10.3.x4/WLS/user_projects/domains/java_
domain/retail/rpm132test/retail-public-security-api/bin> ./update-RMS01USER.sh
usage: update-RMS01USER.sh <username>
<username>: the username to update into this alias.
Example: update-RMS01USER.sh myuser
Note: this script will ask you for the password for the username that you pass in.
mspdev71:[103x4WLS] /u00/webadmin/product/10.3.4/WLS/user_projects/domains/java_
domain/retail/rpm132test/retail-public-security-api/bin>
```

dump_credentials.sh

dump_credentials.sh is used to retrieve information from the wallet. For each entry found in the wallet, the wallet partition, the alias, and the user name are displayed. Note that the password is not displayed. If the value of an entry is uncertain, run save_credential.sh to save the entry with a known password.

```
Dump_credentials.sh <wallet location>
```

Example:

```
dump_credentials.sh /u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_
mck_soa_domain/retail/reim13/config

=============================================
Retail Public Security API Utility
=============================================
```

Below are the credentials found in the wallet at the location: /u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_soa_domain/retail/reim13/config

```
Application level key partition name:reim13
User Name Alias:WLS-ALIAS User Name:weblogic
User Name Alias:RETAIL-ALIAS User Name:retail.user
User Name Alias:LDAP-ALIAS User Name:RETAIL.USER
User Name Alias:RMS-ALIAS User Name:rms132mock
User Name Alias:REIMBAT-ALIAS User Name:reimbat
```

save_credential.sh

save_credential.sh is used to update the information in the wallet. If you are unsure about the information that is currently in the wallet, use dump_credentials.sh as indicated above. You can add new or update using save_credential.sh as shown below:

```
save_credential.sh -a <alias> -u <user> -p <partition name>  -l <path of the
wallet file location where credentials are stored>
```

Example:

```
mspdv351:[103x_WLS] /u00/webadmin/mock132_
testing/rtil/rtil/application/retail-public-security-api/bin> save_credential.sh
-l /u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_soa_
domain/retail/reim13/config
-a RMS-ALIAS -p reim13 -u rms132mock

===========================================
Retail Public Security API Utility
===========================================

Enter password:
Verify password:
```

Note: `-p` in the above command is for partition name. You must specify the proper partition name used in application code for each java application.

`save_credentials.sh` and `dump_credentials.sh` scripts are same for all applications. If using save_credential.sh to add a wallet entry or to update a wallet entry, bounce the application/managed server so that your changes are visible to the application. Also, save a backup copy of your cwallet.sso file in a location outside of the deployment path, because redeployment or reinstallation of the application will wipe the wallet entries you made after installation of the application. To restore your wallet entries after a redeployment/reinstallation, copy the backed up cwallet.sso file over the cwallet.sso file. Then bounce the application/managed server.

Usage:

```
===========================================
Retail Public Security API Utility
===========================================
usage: save_credential.sh -au[plh]
E.g. save_credential.sh -a rms-alias -u rms_user -p rib-rms -l ./
 -a,--userNameAlias <arg>              alias for which the credentials
needs to be stored
 -h,--help                             usage information
 -l,--locationofWalletDir <arg>        location where the wallet file is
created.If not specified, it creates the wallet under secure-credential-wallet
directory which is already present under the retail-public-security-api/
directory.
 -p,--appLevelKeyPartitionName <arg>   application level key partition name
 -u,--userName <arg>                   username to be stored in secure
credential wallet for specified alias*
```

How Does the Wallet Relate to the Application?

The ORACLE Retail Java applications have the wallet alias information you create in an <app-name>.properties file. Below is the reim.properties file. Note the database information and the user are presented as well. The property called datasource.credential.alias=RMS-ALIAS uses the ORACLE wallet with the argument of RMS-ALIAS at the csm.wallet.path and csm.wallet.partition.name = reim13 to retrieve the password for application use.

Reim.properties code sample:

```
datasource.url=jdbc:oracle:thin:@mspdv349.us.oracle.com:1521:pkols07
datasource.schema.owner=rms132mock
datasource.credential.alias=RMS-ALIAS
# ================================================================
# ossa related Configuration
#
# These settings are for ossa configuration to store credentials.
# ================================================================

csm.wallet.path=/u00/webadmin/product/10.3.x/WLS/user_projects/domains/132_mck_
soa_domain/retail/reim13/config
csm.wallet.partition.name=reim13
```

How Does the Wallet Relate to Java Batch Program Use?

Some of the ORACLE Retail Java batch applications have an alias to use when running Java batch programs. For example, alias REIMBAT-ALIAS maps through the wallet to REIM app user reimbat, already on the database. To run a ReIM batch program the format would be: reimbatchpgmname REIMBAT-ALIAS <other arguments as needed by the program in question>.

Setting up RETL Wallets

RETL creates a wallet under `$RFX_HOME/etc/security`, with the following files:

- `cwallet.sso`
- `jazn-data.xml`
- `jps-config.xml`
- `README.txt`

To setup RETL wallets, perform the following steps:

1. Set the following environment variables:
   - `ORACLE_SID=retaildb`
   - `RFX_HOME=/u00/rfx/rfx-13.2.0`
   - `RFX_TMP=/u00/rfx/rfx-13.2.0/tmp`
   - `JAVA_HOME=/usr/jdk1.6.0_12.64bit`
   - `LD_LIBRARY_PATH=$ORACLE_HOME`
   - `PATH=$RFX_HOME/bin:$JAVA_HOME/bin:$PATH`
2. Change directory to `$RFX_HOME/bin`.
3. Run `setup-security-credential.sh`.
   a. Enter 1 to add a new database credential.
   b. Enter the dbuseralias. For example, retl_java_rms01user.
   c. Enter the database username. For example, rms01user.
   d. Enter the database password.

e. Re-enter the database password.

f. Enter D to exit the setup script.

4. Update your RETL environment variable script to reflect the names of both the Oracle Networking wallet and the Java wallet.

   For example, to configure RETLforRPAS, modify the following entries in `$MMHOME/RETLforRPAS/rfx/etc/rmse_rpas_config.env`

   - The RETL_WALLET_ALIAS should point to the Java wallet entry:

     ```
     export RETL_WALLET_ALIAS="retl_java_rms01user"
     ```

   - The ORACLE_WALLET_ALIAS should point to the Oracle network wallet entry:

     ```
     export ORACLE_WALLET_ALIAS="dvols29_rms01user"
     ```

   - export ORACLE_WALLET_ALIAS="dvols29_rms01user"

     ```
     export SQLPLUS_LOGON="/@${ORACLE_WALLET_ALIAS}"
     ```

5. To change a password later, run `setup-security-credential.sh`.

   a. Enter 2 to update a database credential.

   b. Select the credential to update.

   c. Enter the database user to update or change.

   d. Enter the password of the database user.

   e. Re-enter the password.

Quick Guide for Retail Wallets

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RMS batch	DB	<RMS batch install dir (MMHOME)>/.wallet	N/A	<Database SID>_ <Database schema owner>	<rms schema owner>	Compile, execution	Installer	N/A	Alias hard-coded by installer
RMS forms	DB	<forms install dir>/base/.wallet	N/A	<Database SID>_ <Database schema owner>	<rms schema owner>	Compile	Installer	N/A	Alias hard-coded by installer
ARI forms	DB	<forms install dir>/base/.wallet	N/A	<Db_ Ari01>	<ari schema owner>	Compile	Manual	ari-alias	
RMWS forms	DB	<forms install dir>/base/.wallet	N/A	<Database SID>_ <Database schema owner>	<rms schema owner>	Compile forms, execute batch	Installer	N/A	Alias hard-coded by installer
RPM app	DB	<forms install dir>/base/.wallet	N/A	<rms schema owner alias>	<rms schema owner>	Execute batch	Manual	rms-alias	
RWMS auto-log in	JAVA	<forms install dir>/base/.javawallet							
			<RWMS Installation name>	<RWMS database user alias>	< RWMS schema owner>	RWMS forms app to avoid dblogin screen	Installer	rwms 13inst	
			<RWMS Installation name>	BI_ALIAS	<BI Publisher administrative user>	RWMS forms app to connect to BI Publisher	Installer	N/A	Alias hard-coded by installer
AIP app	JAVA	<weblogic domain home>/ retail/ <deployed aip app name>/config							Each alias must be unique
			aip13	<AIP weblogic user alias>	<AIP weblogic user name>	App use	Installer	aip-weblogic-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
			aip13	<AIP database schema user alias>	<AIP data base schema user name>	App use	Installer	aip01user-alias	
			aip13	<rib-aip weblogic user alias>	<rib-aip web logic user name>	App use	Installer	rib-aip-weblogic-alias	
RPM app	JAVA	<weblogic domain home>/ retail/ <deployed rpm app name>/ config							Each alias must be unique
			rpm13	<rpm weblogic user alias>	<rpm weblogic user name>	App use	Installer	rpm-weblogic-alias	
			rpm13	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01 user-alias	
			rpm13	<rpm application user one alias>	<rpm application user one name>	App use	Installer	user1-alias	
			rpm13	<rpm application user two alias>	<rpm application user two name>	App use	Installer	user2-alias	
			rpm13	<rpm batch user alias>	<rpm batch user name>	App, batch use	Installer	rpmbatch-alias	
			rpm13	<rib-rpm weblogic user alias>	<rib-rpm web logic user name>	App use	Installer	rib-rpm-weblogic-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
ReIM app	JAVA	<weblogic domain home>/ retail/ <deployed reim app name>/ config							Each alias must be unique
			<instal led app name>	<reim weblogic user alias>	<reim web logic user name>	App use	Installer	weblogic-alias	
			<instal led app name>	<rms shema user alias>	<rms shema user name>	App, batch use	Installer	rms01 user-alias	
			<instal led app name>	<reim webservice validation user alias>	<reim web service valida tion user name>	App use	Installer	reimweb service-alias	
			<instal led app name>	<reim batch user alias>	<reim batch user name>	App, batch use	Installer	reim batch-alias	
Alloc app	JAVA	<weblogic domain home>/ retail/<dep loyed alloc app name>/ config							Each alias must be unique
			<instal led app name>	<alloc weblogic user alias>	<alloc web logic user name>	App use	Installer	web logic-alias	
			<instal led app name>	<rms schema user alias>	<rms schema user name>	App use	Installer	rms01 user-alias	
			<instal led app name>	<installed app name> <rsl for rms weblogic user alias>	<rsl for rms web logic user name>	App use	Installer	rsl-rms-weblogic-alias	

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
RSL app	JAVA	<RSL INSTALL DIR>/rsl-rms/security/config							Each alias must be unique
			rsl-rsm	<rsl weblogic user alias>	<rsl weblogic user name>	App use	Installer	weblogic-alias	
			rsl-rsm	<rms shema user alias>	<rms shema user name>	App use	Installer	rms01 user-alias	
SIM app	JAVA	<weblogic domain home>/retail/<deployed sim app name> config							
			rpm	<rpm weblogic user alias>	<rpm web logic user name>	App use	Installer	rpm-web logic-alias	
			rms	<rsl for rms weblogic user alias>	<rsl for rms weblogic user name>	App use	Installer	rsl-rms-weblogic-alias	
			rib-sim	<rib-sim weblogic user alias>	<rib-sim weblogic user name>	App use	Installer	rib-sim-weblogic-alias	
RETL	JAVA	<RETL home>/etc/security	N/A	<target application user alias>	<target application db userid>	App use	Manual	retl_java_rms01 user	User may vary depending on RETL flow's target application
RETL	DB	<RETL home>/.wallet	N/A	<target application user alias>	<target application db userid>	App use	Manual	<db>_<user>	User may vary depending on RETL flow's target application
RIB	Java	<RIBHOME DIR>/deployment-home/conf/security							<app> is one of aip, rfm, rms, rpm, sim, rwms, tafr

Retail app	Wallet type	Wallet loc	Wallet partition	Alias name	User name	Use	Create by	Alias Example	Notes
JMS			jms<1-5>	<jms user alias> for jms<1-5>	<jms user name> for jms<1-5>	Integration use	Installer	jms-alias	
Web Logic			rib-<app>#web-app-user-alias	<rib-app weblogic user alias>	<rib-app weblogic user name>	Integration use	Installer	weblogic-alias	
Admin GUI			rib-<app>#web-app-user-alias	<rib-app admin gui user alias>	<rib-app admin gui user name>	Integration use	Installer	admin-gui-alias	
Application			rib-<app>#user-alias	<app weblogic user alias>	<app weblogic user name>	Integration use	Installer	app-user-alias	Valid only for aip, rpm, sim
DB			rib-<app>#app-db-user-alias	<rib-app database schema user alias>	<rib-app database schema user name>	Integration use	Installer	db-user-alias	Valid only for rfm, rms, rwms, tafr
Error Hospital			rib-<app>#hosp-user-alias	<rib-app error hospital database schema user alias>	<rib-app error hospital database schema user name>	Integration use	Installer	hosp-user-alias	

B

Appendix: Installation Order

This appendix provides a guideline for the order in which the Oracle Retail applications should be instaled. If a retailer has chosen to use only some of the applications, the order is still valid, less the applications not being installed.

> **Note:** The installation order is not meant to imply integration between products.

Enterprise Installation Order

1. Oracle Retail Merchandising System (RMS), Oracle Retail Trade Management (RTM), Oracle Retail Sales Audit (ReSA), Optional: Oracle Retail Fiscal Management (ORFM).

   > **Note:** ORFM is an optional application for RMS if you are implementing Brazil localization.

2. Oracle Retail Service Layer (RSL)
3. Oracle Retail Extract, Transform, Load (RETL)
4. Oracle Retail Active Retail Intelligence (ARI)
5. Oracle Retail Warehouse Management System (RWMS)
6. Oracle Retail Price Management (RPM)
7. Oracle Retail Invoice Matching (ReIM)
8. Oracle Retail Allocation

   > **Note:** During installation of RPM, you are asked for the RIBforRPM provider URL. Because RIB is installed after RPM, make a note of the URL you enter. If you need to change the RIBforRPM provider URL after you install RIB, you can do so by editing the remote_service_locator_info_ribserver.xml file.

9. Oracle Retail Central Office (ORCO)
10. Oracle Retail Returns Management (ORRM)
11. Oracle Retail Back Office (ORBO) or Back Office with Labels and Tags (ORLAT)

12. Oracle Retail Store Inventory Management (SIM)

> **Note:** During installation of SIM, you are asked for the RIB provider URL. Because RIB is installed after SIM, make a note of the URL you enter. If you need to change the RIB provider URL after you install RIB, you can do so by editing the remote_service_locator_info_ribserver.xml file.

13. Oracle Retail Predictive Application Server (RPAS)
14. Oracle Retail Demand Forecasting (RDF)
15. Oracle Retail Category Management (CM)
16. Oracle Retail Replenishment Optimization (RO)
17. Oracle Retail Analytic Parameter Calculator Replenishment Optimization (APC RO)
18. Oracle Retail Regular Price Optimzation (RPO)
19. Oracle Retail Merchandise Financial Planning (MFP)
20. Oracle Retail Size Profile Optimization (SPO)
21. Oracle Retail Assortment Planning (AP)
22. Oracle Retail Item Planning (IP)
23. Oracle Retail Item Planning Configured for COE (IP COE)
24. Oracle Retail Advanced Inventory Planning (AIP)
25. Oracle Retail Integration Bus (RIB)
26. Oracle Retail Point-of-Service (ORPOS)
27. Oracle Retail Markdown Optimization (MDO)
28. Oracle Retail Clearance Optimization Engine (COE)
29. Oracle Retail Analytic Parameter Calculator for Markdown Optimization (APC-MDO)
30. Oracle Retail Analytic Parameter Calculator for Regular Price Optimization (APC-RPO)
31. Oracle Retail Promotion Intelligence and Promotion Planning and Optimization (PI-PPO)
32. Oracle Retail Analytics
33. Oracle Retail Workspace (ORW)