Oracle® Retail Merchandising System
Oracle Financials Interface
Implementation Guide Addendum
Release 11.0.8
June 2006

ORACLE

Contents

Preface

This addendum contains updates and corrections to the Oracle Retail RMS – Oracle Financials Implementation Guide.

Audience

Anyone with an interest in developing a deeper understanding of the underlying processes and architecture supporting RMS/Oracle Financials integration functionality will find valuable information in this guide. There are three audiences in general for whom this guide is written:

- Integrators and implementation staff with overall responsibility for implementing RMS and Oracle Financials.
- Business analysts looking for information about processes and interfaces to validate the support for business scenarios within Oracle Financials, RMS, and other systems across the enterprise.
- System analysts and system operations personnel.

Related Documents

You can find more information about this product in these resources:

- Oracle Retail Merchandising System Installation Guide
- Oracle Retail Merchandising System Release Notes
- Oracle Retail Merchandising System Operations Guide
- Oracle Retail Merchandising System Data Model
- Oracle Financials documentation

Customer Support

- https://metalink.oracle.com

When contacting Customer Support, please provide:

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

1
Introduction

The information in this document reflects modifications and updates to the latest Oracle Retail RMS-Oracle Financials Implementation Guide. Each chapter title and section title in this document corresponds to a chapter title and a section title in the Oracle Retail RMS-Oracle Financials Implementation Guide.

Please note that entire sections have been included from the original Oracle Retail RMS-Oracle Financials Implementation Guide for your reference, and the changes that have been made to those sections are in bold or in crossed-out text (signifying deletions).

2
Architecture

Oracle Exchange Rates

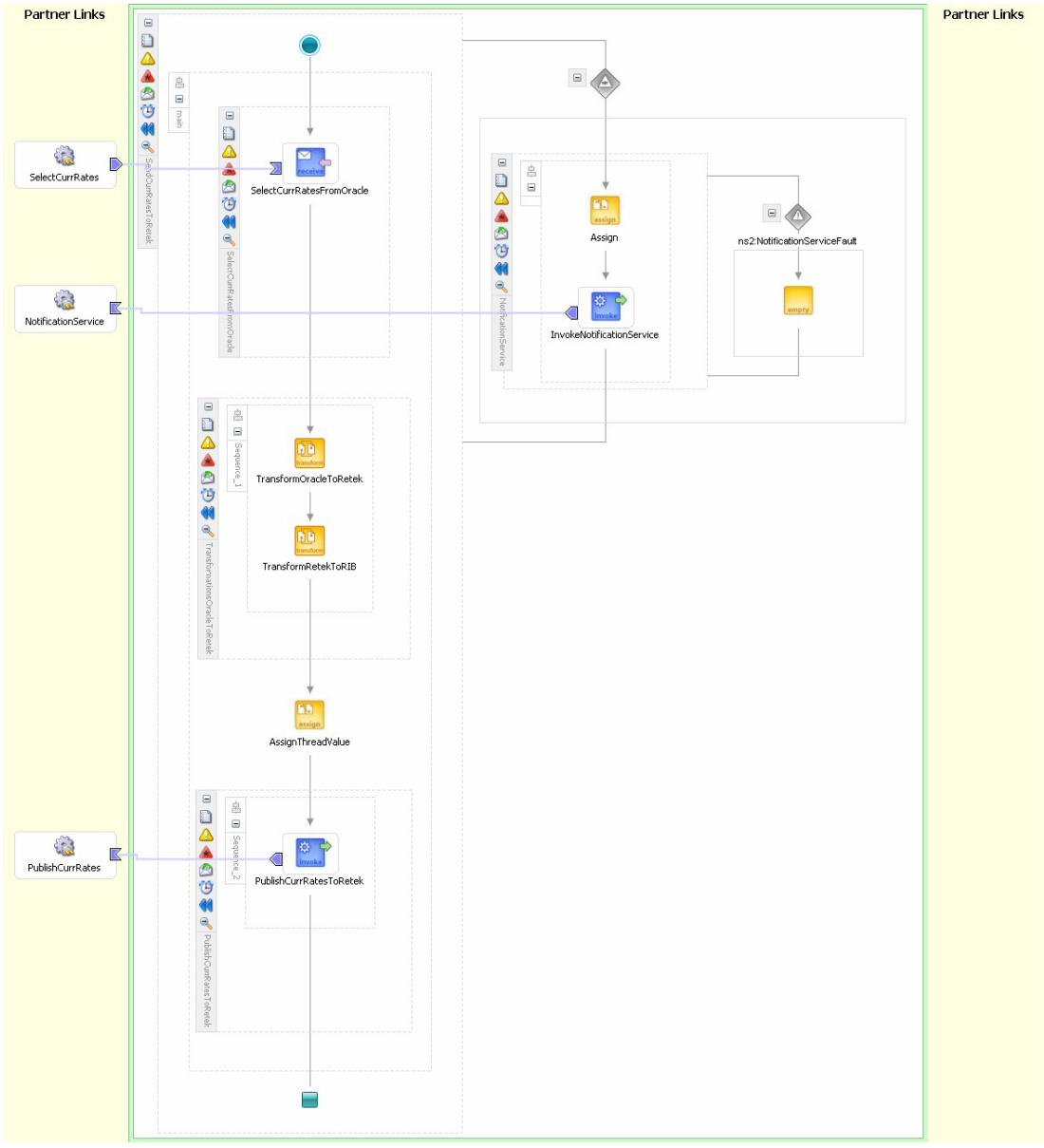
Oracle sends the foreign currency conversion rates information to Oracle Retail. This data is extracted by the BPEL PM and published to the Oracle Retail Integration Bus (RIB) as XML messages on the JMS topic on the RIB. RIB APIs pick these up for insertion into the RMS system. BPEL PM has error handling routines to accommodate system error scenarios while the RIB API's already have error handling routines built-in for errors on their end. These are written to the Error Hospital (error repository) of the RIB.

The design assumes that the following statements are true:

- Oracle Financials and Oracle Retail store currency conversion rates information in different ways. Oracle Financials maintains the rates with respect to a "from currency to currency" on a particular date, whereas Oracle Retail stores only the rates in the context of a "to" currency for a given date. The "from" currency is assumed to always be the RMS primary currency. Oracle Financials sends all conversion rates to Oracle Retail in the message and the Oracle Retail API populates its table with only those currencies where the "from" currency is equal to the RMS primary currency.
- There are different types of currencies used in Oracle Financials and Oracle Retail. Oracle Financials seeds only Spot, Corporate and User rate types as default exchange rate types. Oracle Retail allows for multiple rate types, but only uses one for the default processing – Operational or Consolidation. This is determined based on a system option in RMS. As part of the configuration, a cross-reference has to be created in RMS that allows the data from Oracle Financials to be properly translated into RMS terminology. This cross-reference is held in the FIF_CURRENCY_XREF table in RMS.
- Currency codes (for example, US Dollars) are the same in both Oracle Financials and Oracle Retail. The list of valid currencies is maintained separately via scripts in each application. However, because currency code size is smaller in Oracle Retail (3 chars), currency codes in Oracle Financials must be entered with 3 characters or less.
- For the JNDI entries, in the "oc4j-ra.xml" for the corresponding adapters, the implementation team has to provide the required install specific connection information. The JNDI entries to be modified are:
  - DB Adapter oc4j-ra.xml: Oracle DB connection information.
    - JNDI name: eis/DB/OracleConnection
  - JMS Adapter oc4j-ra.xml: Oracle Retail RIB JMS topic connection information.
    - JNDI name: eis/Jms/SendCurrencyRates
- Provide the value for the 'SYSADMIN_EMAIL' descriptor property from the BPEL console.
- The BPEL requires a starting conversion date as a parameter to the deployment descriptor property "VDate" at installation time. This parameter is used to bind to the SQL statement to select currency conversion rates with conversion dates greater than the Oracle Retail specified setup date Vdate.

- **It is recommended that exchange rates be entered individually for each date. Exchange rates entered through Rates Manager, Date range functionality, or the interface table are date stamped, but not time stamped. This means the creation/update date stamp is less than the last time BPEL polled. Therefore, these records would be ignored.**

SendCurrRatesToOracle Retail BPEL Process



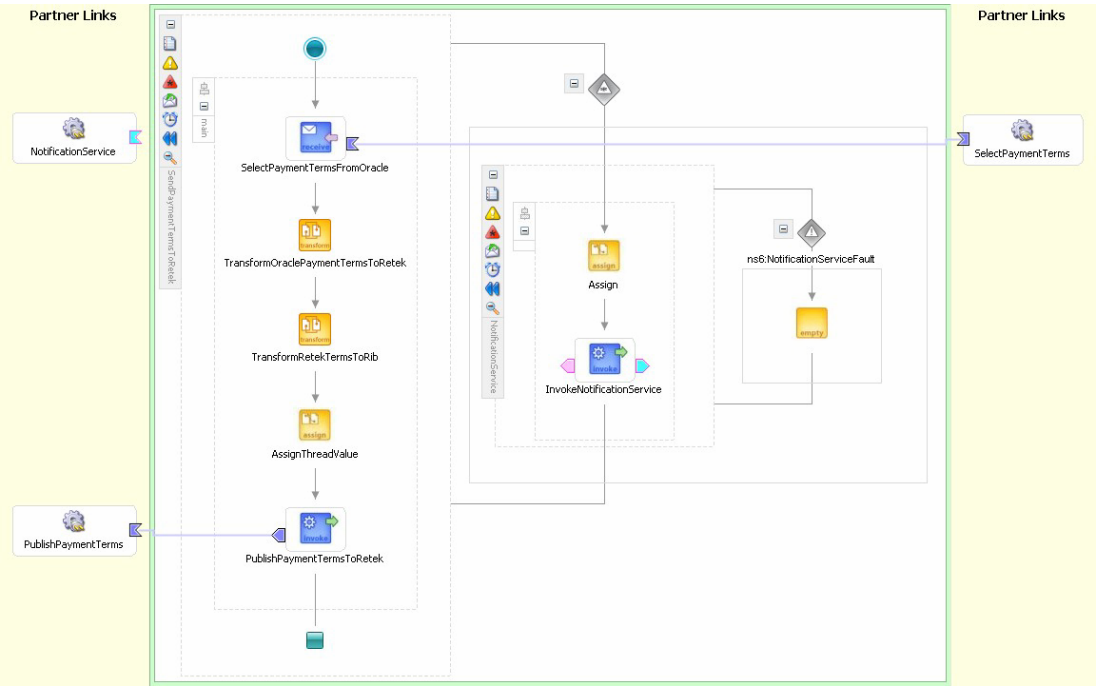
Oracle Payment Terms

Oracle sends the payment terms reference data to Oracle Retail. This data is extracted by the BPEL processing manager and published to the Oracle Retail Integration Bus (RIB) as XML messages on the JMS Topic on the RIB. RIB APIs then pick these up for insertion into the RMS system. BPEL PM has error handling routines to accommodate system error scenarios while the RIB APIs already have error handling routines built in for errors on their end. These are written to the Error Hospital (error repository) of the RIB.

This design assumes that the following statements are true:

- Oracle only interfaces payment terms in the base installed language to Oracle Retail. The other language translations are handled in Oracle Retail, using base translation functionality.
- Payment Terms Rank information is not mandatory in Oracle Financials. It is a required attribute in Oracle Retail. Oracle Financials default a value of '1' for the Rank for any payment terms that has no rank specified in Oracle Financials.
- Payment Terms name is used for Payment Terms Description when the Description field is not populated in Oracle.
- Payment Terms Enabled flag value is evaluated based on the start and end effective dates of the payment terms and the current system date.
- The Due Days, Due Amount, Discount Days, Discount Percent, Discount Months Forward are defaulted to '0' when it is not populated in Oracle Financials.
- The Due Day of the Month, Discount Day of the Month and Due Cut Off Day are defaulted to "1" when not populated in Oracle Financials.
- **The discount calculated by Oracle Retail may be different than the discount calculated by Oracle Payables due to differences in data models.**

SendPaymentTermsToOracle Retail BPEL Process



Stock Ledger, Invoices and Sales to GL

Oracle Retail populates the STG_FIF_GL_DATA staging table with GL transactional data related to Oracle Retail's Sales Audit system and stock ledger system. For information about how Oracle Retail engages in this processing, see the latest Oracle Retail Invoice Matching (ReIM) Operations Guide and/or Operations Guide Addendum.

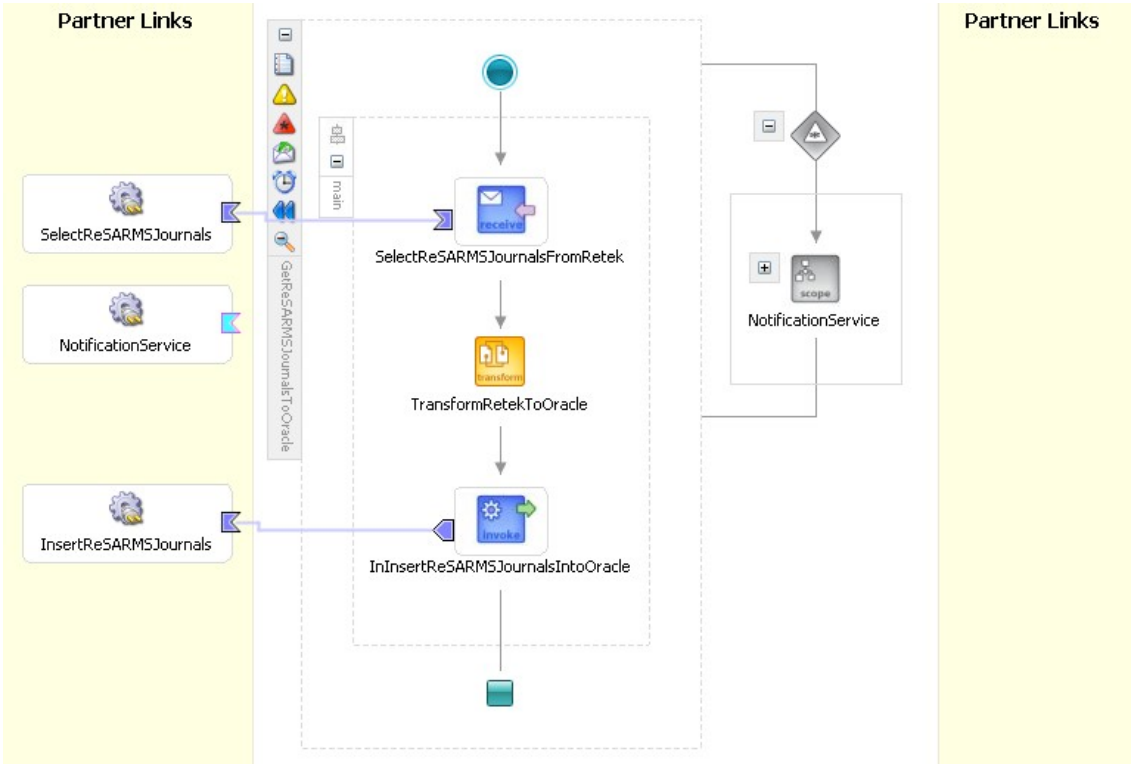
Oracle pulls this GL data from the Oracle Retail staging table, performs appropriate transformations and populates the GL_INTERFACE table in Oracle E-Business Suite. This process is orchestrated by BPEL PM using DB Adapters.

BPEL PM has error handling routines to accommodate system error scenarios. If an error is encountered, a notification is sent to a designated System Administrator role with pertinent information related to the BPEL process that has encountered the error. The System Administrator uses the built-in BPEL console to trouble-shoot and re-process the data.

This design assumes that the following statements are true:

- The Oracle Retail batch process that populates STG_FIF_GL_DATA populates the complete dataset for a given period (daily, monthly, and so on). Hence, there are **no** checks and balances to pre-validate if the journals are balanced, prior to processing the records.
- The GL inbound process designed by Oracle is responsible for archiving the data in STG_FIF_GL_DATA to a history table after successful processing has occurred.
- Invoking of the standard concurrent program 'Journal Import' is a manual process and the modality of invoking this is an implementation-time decision and does not form part of this design.
- This design has a dependency on an agreed upon data mapping between the Oracle Retail staging table, STG_FIF_GL_DATA, and the GL_INTERFACE table on the Oracle side. This data mapping includes any required transformation and application of business rules to ensure data integrity in Oracle General Ledger. Please refer to the Appendix for the detailed Data Mapping and transformations.
- The design assumes that Oracle Retail always populates the code_combinations_id, and does not send the accounting segment information. (The accounting segment information is still populated on the table. However, Oracle does not use this information).
- The following master data synchronization and setup/configuration must be completed before running this interface:
  - Chart of Accounts synchronization between Oracle (Master) and Oracle Retail (Slave).
  - Currency Codes and rates synchronization between Oracle (Master) and Oracle Retail (Slave).
  - Oracle Retail Journal Source and Category have been set up in Oracle General Ledger.
  - Currency Conversion Type. Only 'Corporate' and 'Spot' conversion types would be supported by this design out-of-box.
  - Period Names and Accounting Calendar synchronization between Oracle (Master) and Oracle Retail (Slave).
  - Only 'Actual' Journals are interfaced and the design does **not** handle Budget and Encumbrance Journals.
- **As part of setup, the following journal sources and journal categories must be created manually:**
  - **Journal sources of 'RETEK' and 'Retail Invoices'.**
  - **Journal categories of 'RETEK', 'Writeoffs', 'Prepayments', and 'Manual Payments'.**
- **Retail.FIF_GL_SETUP.PERIOD_NAME needs to be populated with the relevant accounting period from Oracle General Ledger Calendar. This must be maintained as a manual business process.**

SendReSAToOracle BPEL Process



Oracle BPEL Product Development has provided a patch to enable Batch Delete and Batch Commit for DB Adapter polling service. Apply this on the BPEL server on which the GL Inbound BPEL process is deployed by executing the following code.

```
 java -jar
D:\bpm_ga1\integration\orabpel\system\appserver\oc4j\j2ee\home\admin.jar
ormi://localhost/ admin welcome -deployconnector -file DbAdapter.rar -name
DbAdapter
```

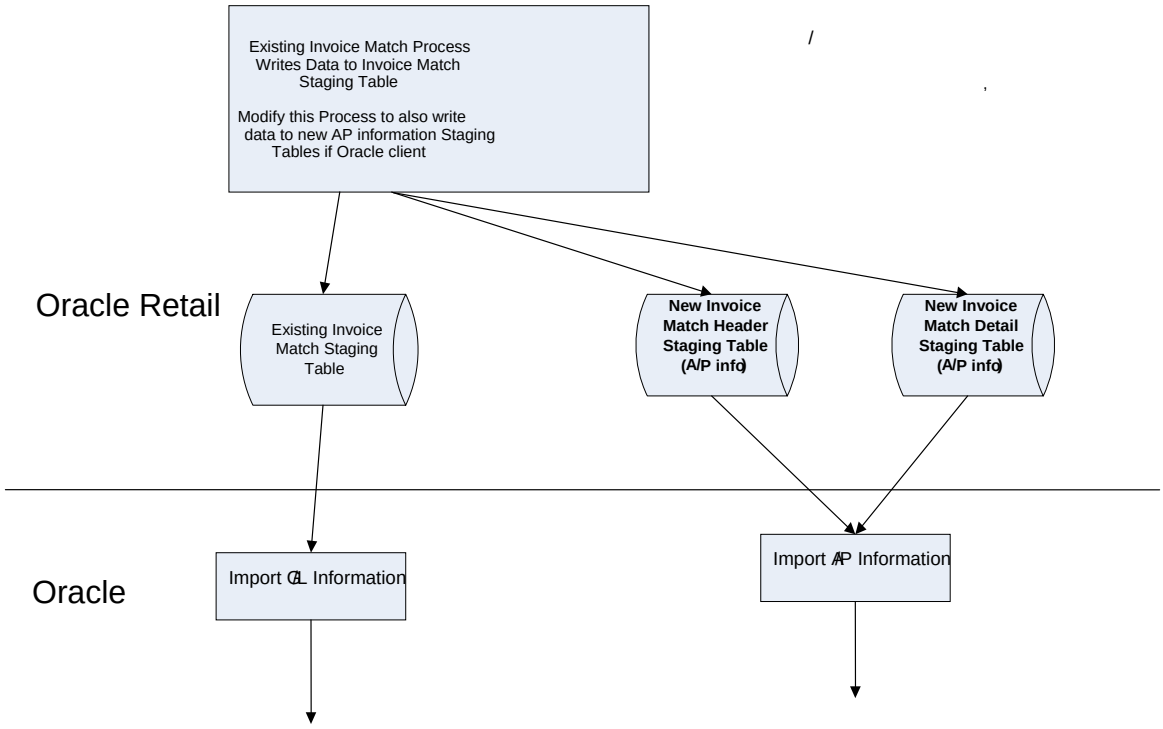
Run this command from the location where you have downloaded the .rar file. This needs to be done only if the Oracle BPEL PM version does not include this patch.

ReIM Approved Payments to AP

Full Invoice Match processing is completed in ReIM with payment recommendations imported to Oracle Accounts Payable. ReIM matches merchandise receipts with merchandise invoices, performing automated and manual matching and discrepancy resolution processing. Matched invoices are posted to interface staging tables specifying the amount ~~and date to pay~~, vendor, Oracle Site ID, GL COA information, and payment terms. Other payables documents, including Debit Memos, Credit Memos and Credit Notes are also interfaced to Oracle Payables via the ReIM staging tables (IM_AP_STAGE_HEAD and IM_AP_STAGE_DETAIL). For information about how Oracle Retail engages in this processing, see the latest Oracle Retail Invoice Matching (ReIM) Operations Guide and/or Operations Guide Addendum.

Certain transactions from ReIM are not interfaced to Oracle Payables, but instead are interfaced to Oracle General Ledger via the IM_FINANCIAL_STAGE table.

The Oracle Retail financial staging table IM_FINANCIALS_STAGE, which stores the GL transactions from Oracle Retail's Invoice Match System, is the primary starting point for the BPEL import process to Oracle. In order to support the transformation and population into Oracle Account Payables standard interface payables transactions are written to a new set of Oracle Retail header/detail staging tables for the Oracle BPEL process - IM_AP_STAGE_HEAD and IM_AP_STAGE_DETAIL.

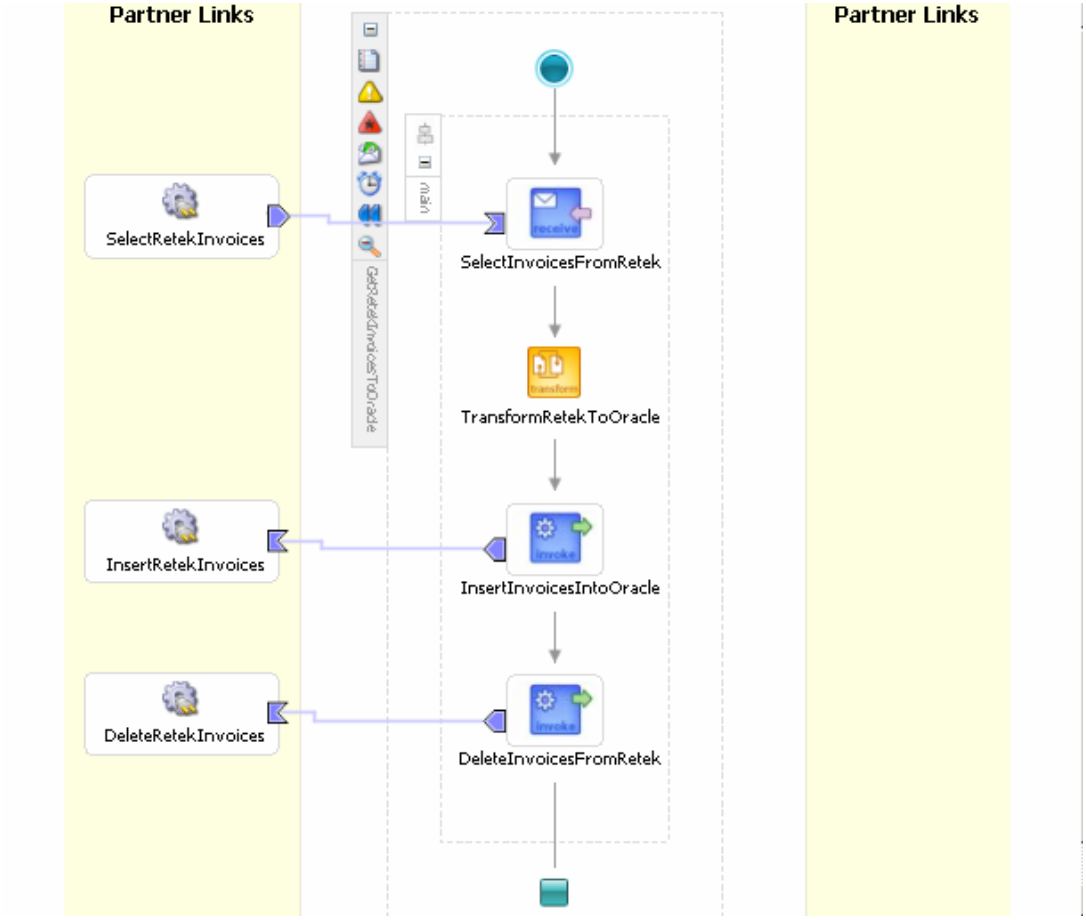


The payables open interface import, which currently accepts only liability account's code_combination_id at the invoice header level, has been enhanced to also accept the concatenated segments as input and generates the code_combination_id accordingly and is stamped on the imported invoice header record in ap_invoices.

This design assumes that the following statements are true:

- The Oracle Retail staging tables (IM_FINANCIALS_STAGE, IM_AP_STAGE_HEAD and IM_AP_STAGE_DETAIL tables) are populated by an Oracle Retail batch process. This batch process populates the complete dataset for the period it is run, which is typically daily.
- Chart of Accounts synchronization between Oracle and Oracle Retail takes place via a separate asynchronous process. It is ensured that the synchronization is performed as an offline business process, and is not in the scope of this design. A designated GL administrator will handle any omissions or errors related to GL code combinations manually.
- GL Cross reference mapping is set up in ReIM by transaction type. This setup process allows for dynamic mapping to different GL COA values by RMS location and RMS merchandise hierarchy information. When ReIM is implemented with Oracle Financials, the Trade Accounts Payable transaction codes in ReIM are not dynamically mapped to GL COA values, but the Oracle Payables Flexfield functionality allows for similar functionality to the ReIM dynamic mapping process based on the dynamic mapping of details lines in the interface tables.
- In the case of importing invoices from ReIM to payables interface tables, Oracle Retail's staging tables populate the segment information for the accounting flex-fields for both header and detail staging tables. Note, however, that the payables interface tables accept the concatenated segments for these flex-fields. This BPEL process assumes that "-" (hyphen) is the delimiter for the chart of accounts implementation on the Oracle Applications.
- GetReIMJournalsToOracle BPEL process assumes that VAT codes in Oracle Retail invoice matching system and the Oracle applications are the same. These codes need to be manually synchronized.
- All the outbound interface processes that interface the setup data from Oracle Financials Applications to Oracle Retail are run before running the Inbound Transaction Interface processes.

GetOracle RetailInvoicesToOracle (Inbound ReIM Invoices)



BPEL Processing and Scheduling

The BPEL Processes are scheduled to occur at defined intervals based on volume and business practice. The database adapter keeps polling the source Oracle database at predetermined time intervals for new or changed records for the outbound reference data processes. **The defined time interval is five minutes.**

Once the BPEL process is initiated it looks for create/update records in the appropriate tables/views in Oracle to select only the records with the last_update_date greater that that last processing run.

For example:

> The adapter uses a Control (Sequencing) table AP_SEQUENCING_HELPER to refer to the LAST_UPDATE_DATE for each process.

The GL COA process looks at “GL_CODE_COMBINATIONS” for create/update. After the selection and raising of the records to the BPEL instances, the LAST_READ_DATE in AP_SEQUENCING_HELPER is updated to the latest value processed. Selected records are processed in the BPEL flow and published to the Oracle Retail JMS for subscription by the RIB. The RIB has been equally configured

3

Oracle Retail Integration Bus (RIB)

For detailed instructions on the installation and operation of the Oracle Retail Integration Bus, refer to the RIB 11.1 Installation Guide, RIB 11.1 Technical Architecture Guide, and RIB 11.1 Operations Guide.

Subscribing Adapter Overview

Subscribing adapters are responsible for ensuring that messages are processed in the correct sequence for a given business entity. For example, for a specific ~~Purchase Order~~ **Matched Invoice**, its "Create ~~Purchase Order~~ **Matched Invoice (MI)**" message must always be processed before an update or delete message. Furthermore, all updates must be processed in the correct order to ensure that two systems are correctly synchronized. But no such guarantee exists when comparing messages concerning different business entities. If no errors occur, messages are processed in a First-In, First-Out (FIFO) order. Alternatively, if an error occurs processing a message for one business object (~~PO~~ **MI** #123), then other messages that apply to other business objects (~~PO~~ **MI**'s #124, #125…) should still be processed. Furthermore, all messages for the problem business object (~~PO~~ **MI** #123) are held in the Error Hospital. If an error occurs during message processing a two-step process is followed:

1. The subscribing adapter notes this internally (**not** in the database) and rolls back all database work associated with the message.
2. The JMS server re-sends the message to the adapter and since it has yet to be processed successfully, the adapter recognizes this message is problematic (sick) and checks it into an Error Hospital database.

A subscribing adapter always checks the hospital database to see if there are any messages in the hospital that act on the same business entity (such as a ~~PO~~ **MI**) that the current message does. If so, the adapter immediately places the current message in the hospital as well. This is to ensure that all messages for a given business entity are processed in the correct order. Without manual intervention, the RIB always processes the "sick" messages for a business object before any subsequent messages that act on the same business object.

After a message is checked into the Error Hospital, a hospital retry adapter/e*Way/Daemon is used to re-post the message to the JMS in order to retry its processing. The assumption is that the error is a transitory one – records locked or there is an external dependency that has not been met. The number of times a message is retried is configurable.

A
Appendix

Supplier

field passed from Oracle	>	Oracle Retail RIB - VendorHdrDesc	API Create	API Update	>	Oracle Retail Database - SUPS	window edit (supvwedt)	comments or special info
~~vendor id~~ **Segment 1**		SUPPLIER	(key)	(key)		SUPPLIER	(key)	
vendor name		SUP_NAME	pass	pass		SUP_NAME	disable	
(will send one occ)		CONTACT_NAME	pass	ignore		CONTACT_NAME	normal	~~Oracle keeps info at lower level.~~ **Oracle Financials stores this data at either the supplier or supplier site level.** They will send info from one occurrence which will be used in Oracle Retail for create only. After that, any updates will be done by a user via Oracle Retail window.
(will send one occ)		CONTACT_PHONE	pass	ignore		CONTACT_PHONE	normal	see note for contact_name above
(will send one occ)		CONTACT_FAX	pass	ignore		CONTACT_FAX	normal	see note for contact_name above

field passed from Oracle	>	Oracle Retail RIB - VendorHdrDesc	API Create	API Update	>	Oracle Retail Database - SUPS	window edit (supvwedt)	comments or special info
---		CONTACT_PAGER	default	ignore		CONTACT_PAGER	normal	
(derived)		SUP_STATUS	see note	see note		SUP_STATUS	see note	For API: see info in Functional Spec. For window: Disable this field EXCEPT if the client uses VAT and current Status is inactive. In that case, if user tries to change status to active, invoke standard edit that VAT region must be valued in order to change status to active.
---		QC_IND	default	ignore		QC_IND	normal	
---		QC_PCT	default	ignore		QC_PCT	normal	
---		QC_FREQ	default	ignore		QC_FREQ	normal	
---		VC_IND	default	ignore		VC_IND	normal	
---		VC_PCT	default	ignore		VC_PCT	normal	
---		VC_FREQ	default	ignore		VC_FREQ	normal	
(will send one occ)		CURRENCY_CODE	pass	ignore		CURRENCY_CODE	normal	see note for contact_name above
(will send one occ)		LANG	pass	ignore		LANG	normal	see note for contact_name above.

field passed from Oracle	>	Oracle Retail RIB - VendorHdrDesc	API Create	API Update	>	Oracle Retail Database - SUPS	window edit (supvwedt)	comments or special info
(will send one occ)		TERMS	pass	ignore		TERMS	normal	see note for contact_name above
(will send one occ)		FREIGHT_TERMS	pass	ignore		FREIGHT_TERMS	normal	see note for contact_name above
---		RET_ALLOW_IND	default	ignore		RET_ALLOW_IND	normal	
---		RET_AUTH_REQ	default	ignore		RET_AUTH_REQ	normal	
---		RET_MIN_DOL_AMT	default	ignore		RET_MIN_DOL_AMT	normal	
---		RET_COURIER	default	ignore		RET_COURIER	normal	
---		HANDLING_PCT	default	ignore		HANDLING_PCT	normal	
---		EDI_PO_IND	default	ignore		EDI_PO_IND	normal	
---		EDI_PO_CHG	default	ignore		EDI_PO_CHG	normal	
---		EDI_PO_CONFIRM	default	ignore		EDI_PO_CONFIRM	normal	
---		EDI_ASN	default	ignore		EDI_ASN	normal	
---		EDI_SALES_RPT_FREQ	default	ignore		EDI_SALES_RPT_FREQ	normal	
---		EDI_SUPP_AVAILABLE_IND	default	ignore		EDI_SUPP_AVAILABLE_IND	normal	
---		EDI_CONTRACT_IND	default	ignore		EDI_CONTRACT_IND	normal	
---		EDI_INVC_IND	default	ignore		EDI_INVC_IND	normal	
---			default	n/a		EDI_CHANNEL_ID	normal	
---		COST_CHG_PCT_VAR	default	ignore		COST_CHG_PCT_VAR	normal	
---		COST_CHG_AMT_VAR	default	ignore		COST_CHG_AMT_VAR	normal	

field passed from Oracle	>	Oracle Retail RIB - VendorHdrDesc	API Create	API Update	>	Oracle Retail Database - SUPS	window edit (supvwedt)	comments or special info
---		REPLEN_APPROVAL_IND	default	ignore		REPLEN_APPROVAL_IND	normal	
---		SHIP_METHOD	default	ignore		SHIP_METHOD	normal	
---		PAYMENT_METHOD	default	ignore		PAYMENT_METHOD	normal	
---		CONTACT_TELEX	default	ignore		CONTACT_TELEX	normal	
(will send one occ)		CONTACT_EMAIL	pass	ignore		CONTACT_EMAIL	normal	see note for contact_name above
---		SETTLEMENT_CODE	default	ignore		SETTLEMENT_CODE	normal	
---		PRE_MARK_IND	default	ignore		PRE_MARK_IND	normal	
---		AUTO_APPR_INVC_IND	default	ignore		AUTO_APPR_INVC_IND	normal	
---		DBT_MEMO_CODE	default	ignore		DBT_MEMO_CODE	normal	
---		FREIGHT_CHARGE_IND	default	ignore		FREIGHT_CHARGE_IND	normal	
---		AUTO_APPR_DBT_MEMO_IND	default	ignore		AUTO_APPR_DBT_MEMO_IND	normal	
---		PREPAY_INVC_IND	default	ignore		PREPAY_INVC_IND	normal	
---		BACKORDER_IND	default	ignore		BACKORDER_IND	normal	
see note		VAT_REGION	See note	ignore		VAT_REGION	normal	For API: see info in Functional Spec.
---		INV_MGMT_LVL	default	ignore		INV_MGMT_LVL	normal	
---		SERVICE_PERF_REQ_IND	default	ignore		SERVICE_PERF_REQ_IND	normal	

field passed from Oracle	>	Oracle Retail RIB - VendorHdrDesc	API Create	API Update	>	Oracle Retail Database - SUPS	window edit (supvwedt)	comments or special info
---		INVC_PAY_LOC	default	ignore		INVC_PAY_LOC	normal	
---		INVC_RECEIVE_LOC	default	ignore		INVC_RECEIVE_LOC	normal	
---		ADDINVC_GROSS_NET	default	ignore		ADDINVC_GROSS_NET	normal	
---		DELIVERY_POLICY	default	ignore		DELIVERY_POLICY	normal	
---		COMMENT_DESC	default	ignore		COMMENT_DESC	normal	
---		DEFAULT_ITEM_LEAD_TIME	default	ignore		DEFAULT_ITEM_LEAD_TIME	normal	
---		DUNS_NUMBER	default	ignore		DUNS_NUMBER	normal	
---		DUNS_LOC	default	ignore		DUNS_LOC	normal	
---		BRACKET_COSTING_IND	default	ignore		BRACKET_COSTING_IND	normal	
---		VMI_ORDER_STATUS	default	ignore		VMI_ORDER_STATUS	normal	
---		DSD_SUPPLIER_IND	default	ignore		DSD_IND	normal	
---		END_DATE_ACTIVE	ignore	ignore		n/a	n/a	

Addresses

Note: Records in Oracle Financials will be sent to Oracle Retail when the following fields are populated in Oracle Financials: address_line1, city, and zip.

field passed from Oracle	>	Oracle Retail RIB - VendorAddrDesc	API Create	API Update	>	Oracle Retail Database - ADDR	window edit (addr)	comments or special info
			(key)	**(key)**		**ADDR_KEY**	n/a	
(literal)		**MODULE**	**(key)**	**(key)**		**MODULE**	n/a	should always be the literal 'SUPP'
vendor_id		**KEY_VALUE_1**	**(key)**	**(key)**		**KEY_VALUE_1**	n/a	this is the supplier number
null		KEY_VALUE_2	default	ignore		KEY_VALUE_2	n/a	
null		SEQ_NO	generate	ignore		SEQ_NO	n/a	
?		**ADDR_TYPE**	**see note**	**see note**		**ADDR_TYPE**	n/a	
null		PRIMARY_ADDR_IND	generate	generate		PRIMARY_ADDR_IND	normal	logic in API will determine
address_line1		**ADD_1**	**pass**	**pass**		**ADD_1**	**disable**	
address_line2		**ADD_2**	**pass**	**pass**		**ADD_2**	**disable**	
address_line3		**ADD_3**	**pass**	**pass**		**ADD_3**	**disable**	
city		**CITY**	**pass**	**pass**		**CITY**	**disable**	
		STATE				**STATE**	**disable**	
		COUNTRY_ID				COUNTRY_ID	disable	
zip		**POST**	**pass**	**pass**		**POST**	**disable**	

field passed from Oracle	>	Oracle Retail RIB - VendorAddrDesc	API Create	API Update	>	Oracle Retail Database - ADDR	window edit (addr)	comments or special info
(will send one occ)		**CONTACT_NAME**	**pass**	ignore		**CONTACT_NAME**	normal	~~Oracle keeps information at lower level. They will send information from one occurrence which will be used in Oracle Retail for create only. After that, any updates will be done by user via Oracle Retail window.~~ **Oracle Financials will pass the first contact name for each address created.**
(will send one occ)		**CONTACT_PHONE**	**pass**	ignore		**CONTACT_PHONE**	normal	see note for contact_name above
---		CONTACT_TELEX	default	ignore		CONTACT_TELEX	normal	
(will send one occ)		**CONTACT_FAX**	**pass**	ignore		**CONTACT_FAX**	normal	see note for contact_name above
(will send one occ)		**CONTACT_EMAIL**	**pass**	ignore		**CONTACT_EMAIL**	normal	see note for contact_name above
null		ORACLE_VENDOR_SITE_ID	default	ignore		ORACLE_VENDOR_SITE_ID	n/a	
n/a		n/a	default	n/a		EDI_ADDR_CHG	n/a	
n/a		n/a	default	n/a		COUNTY	n/a	

field passed from Oracle	>	Oracle Retail RIB - VendorAddrDesc	API Create	API Update	>	Oracle Retail Database - ADDR	window edit (addr)	comments or special info
n/a		n/a	default	n/a		PUBLISH_IND	n/a	
Mutliple occurrences of info at lower level:								
org_id		**oracle_org_unit_id**	**pass**	**pass**		**oracle_org_unit_id**	n/a	
vendor_site_id		**oracle_vendor_site_id**	**pass**	**pass**		**oracle_vendor_site_id**	n/a	

Oracle Retail Staging Table FIF_STG_GL_DATA

Column Name`	Data Type	Mandatory ?		Column Name	Data Type	Mandatory	Remarks
STATUS	VARCHAR2(50)			STATUS	VARCHAR2(50)	Yes	Straight Mapping.
SET_OF_BOOKS_ID	NUMBER(15)			SET_OF_BOOKS_ID	NUMBER(15)	Yes	Straight Mapping. Oracle Retail will send valid value.
ACCOUNTING_DATE	DATE			ACCOUNTING_DATE	DATE	Yes	Straight Mapping.
CURRENCY_CODE	VARCHAR2(15)			CURRENCY_CODE	VARCHAR2(15)	Yes	Straight Mapping. Oracle Retail will synchronize reference data with Oracle.
DATE_CREATED	DATE			DATE_CREATED	DATE	Yes	Straight Mapping.
CREATED_BY	NUMBER(15)			CREATED_BY	NUMBER(15)	Yes	
ACTUAL_FLAG	VARCHAR2(1)			ACTUAL_FLAG	VARCHAR2(1)	Yes	Straight Mapping.
USER_JE_CATEGORY_NAME	VARCHAR2(25)			USER_JE_CATEGORY_NAME	VARCHAR2(25)	Yes	Straight Mapping. Oracle Retail will send valid value.
USER_JE_SOURCE_NAME	VARCHAR2(25)			USER_JE_SOURCE_NAME	VARCHAR2(25)	Yes	Straight Mapping. Oracle Retail will send valid value.
CURRENCY_CONVERSION_DATE	DATE			CURRENCY_CONVERSION_DATE	DATE		Straight Mapping.
CURRENCY_CONVERSION_TYPE	VARCHAR2(30)			USER_CURRENCY_CONVERSION_TYPE	VARCHAR2(30)		Only Valid Values are ‘Corporate’ and’Spot’.

Column Name`	Data Type	Mandatory ?		Column Name	Data Type	Mandatory	Remarks
				CURRENCY_CONVERSION_RATE	NUMBER		Will not be sent by Oracle Retail. Conversion Rate will be derived from GL_DAILY_RATES table based on conversion date, conversion type, currency_code and functional currency.
ACCT_SEGMENT1	VARCHAR2(25)			SEGMENT1	VARCHAR2(25)		Will not be sent by Oracle Retail as the Code_Combination_id would be sent.
ACCT_SEGMENT2	VARCHAR2(25)			SEGMENT2	VARCHAR2(25)		Will not be sent by Oracle Retail as the Code_Combination_id would be sent.
ACCT_SEGMENT3	VARCHAR2(25)			SEGMENT3	VARCHAR2(25)		Will not be sent by Oracle Retail as the Code_Combination_id would be sent.
ACCT_SEGMENT4	VARCHAR2(25)			SEGMENT4	VARCHAR2(25)		Will not be sent by Oracle Retail as the Code_Combination_id would be sent.
ACCT_SEGMENT5	VARCHAR2(25)			SEGMENT5	VARCHAR2(25)		Will not be sent by Oracle Retail as the Code_Combination_id would be sent.

Column Name`	Data Type	Mandatory?		Column Name	Data Type	Mandatory	Remarks
ACCT_SEGMENT6	VARCHAR2(25)			SEGMENT6	VARCHAR2(25)		Will not be sent by Oracle Retail as the Code_Combination_id would be sent.
ACCT_SEGMENT7	VARCHAR2(25)			SEGMENT7	VARCHAR2(25)		Will not be sent by Oracle Retail as the Code_Combination_id would be sent.
ACCT_SEGMENT8	VARCHAR2(25)			SEGMENT8	VARCHAR2(25)		Will not be sent by Oracle Retail as the Code_Combination_id would be sent.
ACCT_SEGMENT9	VARCHAR2(25)			SEGMENT9	VARCHAR2(25)		Will not be sent by Oracle Retail as the Code_Combination_id would be sent.
ACCT_SEGMENT10	VARCHAR2(25)			SEGMENT10	VARCHAR2(25)		Will not be sent by Oracle Retail as the Code_Combination_id would be sent.
ENTERED_DR_AMOUNT	NUMBER(20,4)			ENTERED_DR	NUMBER		Straight Mapping and truncation per Oracle Retail column sizing constraints.
ENTERED_CR_AMOUNT	NUMBER(20,4)			ENTERED_CR	NUMBER		Straight Mapping and truncation per Oracle Retail column sizing constraints.

Column Name`	Data Type	Mandatory ?		Column Name	Data Type	Mandatory	Remarks
				ACCOUNTED_DR	NUMBER		Will not be sent. This will be derived during GL Import rom the GL Daily rates table.
				ACCOUNTED_CR	NUMBER		Will not be sent. This will be derived during GL Import rom the GL Daily rates table.
TRANSACTION_DATE	DATE			TRANSACTION_DATE	DATE		Straight Mapping.
REFERENCE1	VARCHAR2(20)			REFERENCE21	VARCHAR2(100)		Straight Mapping.
REFERENCE2	VARCHAR2(20)			REFERENCE22	VARCHAR2(240)		Straight Mapping.
REFERENCE3	VARCHAR2(20)			REFERENCE23	VARCHAR2(100)		Straight Mapping.
REFERENCE4	VARCHAR2(20)			REFERENCE24	VARCHAR2(100)		Straight Mapping.
REFERENCE5	VARCHAR2(20)			REFERENCE25	VARCHAR2(240)		Straight Mapping.
				JE_BATCH_ID	NUMBER(15)		Do Not Map

Column Name`	Data Type	Mandatory ?		Column Name	Data Type	Mandatory	Remarks
PERIOD_NAME	VARCHAR2(15			PERIOD_NAME	VARCHAR2(15)		Straight Mapping. **The Period Names need to be populated with the relevant accounting period from Oracle General Ledger Calendar. This must be manually synchronized.**
				JE_HEADER_ID	NUMBER(15)		Leave this Null. For Internal Use only
				JE_LINE_NUM	NUMBER(15)		Do Not Map
				CHART_OF_ACCOUNTS_ID	NUMBER(15)		Do Not Map
				FUNCTIONAL_CURRENCY_CODE	VARCHAR2(15)		Do Not Map
CODE_COMBINATION_ID	NUMBER(15)			CODE_COMBINATION_ID	NUMBER(15)		Straight Mappig.
ATTRIBUTE1	VARCHAR2(20)			ATTRIBUTE1	VARCHAR2(150)		Do not pull these columns
ATTRIBUTE2	VARCHAR2(20)			ATTRIBUTE2	VARCHAR2(150)		Do not pull these columns
				GL_SL_LINK_ID	NUMBER		Leave these null
				GL_SL_LINK_TABLE	VARCHAR2(30)		Leave these null

Column Name`	Data Type	Mandatory ?		Column Name	Data Type	Mandatory	Remarks
ATTRIBUTE3	VARCHAR2(20)			ATTRIBUTE3	VARCHAR2(150)		Do not pull these columns
ATTRIBUTE4	VARCHAR2(20)			ATTRIBUTE4	VARCHAR2(150)		Do not pull these columns
ATTRIBUTE5	VARCHAR2(20)			ATTRIBUTE5	VARCHAR2(150)		Do not pull these columns
ATTRIBUTE6	VARCHAR2(20)			ATTRIBUTE6	VARCHAR2(150)		Do not pull these columns
				CONTEXT	VARCHAR2(150)		Do not pull these columns
				CONTEXT2	VARCHAR2(150)		Do not pull these columns
				INVOICE_DATE	DATE		Do Not Map.
				TAX_CODE	VARCHAR2(15)		Do Not Map.
				INVOICE_IDENTIFIER	VARCHAR2(20)		Do Not Map.
				INVOICE_AMOUNT	NUMBER		Do Not Map.
				CONTEXT3	VARCHAR2(150)		Do Not Map.
				USSGL_TRANSACTION_CODE	VARCHAR2(30)		Do Not Map.
				DESCR_FLEX_ERROR_MESSAGE	VARCHAR2(240)		Do Not Map.
				JGZZ_RECON_REF	VARCHAR2(240)		Do Not Map.

Column Name`	Data Type	Mandatory?		Column Name	Data Type	Mandatory	Remarks
				REFERENCE_DATE	DATE		Do Not Map.
PGM_NAME	VARCHAR2(100)	No					Do Not Map.
				ENCUMBRANCE_TYPE_ID	NUMBER		Do Not Map
				BUDGET_VERSION_ID	NUMBER		Do Not Map
				AVERAGE_JOURNAL_FLAG	VARCHAR2(1)		Do Not Map
				ORIGINATING_BAL_SEG_VALUE	VARCHAR2		Do Not Map
				SEGMENT11-30	VARCHAR2(25)		Do Not Map
				REFERENCE1,3,4,6-9	VARCHAR2(100)		Do Not Map
				REFERENCE10,2,5	VARCHAR2(240)		Do Not Map
				REFERENCE11-20	VARCHAR2(100)		Do Not Map
				REFERENCE26-30	VARCHAR2(240)		Do Not Map
				DATE_CREATED_IN_GL	DATE		Do Not Map
				WARNING_CODE	VARCHAR2(4)		Do Not Map

Column Name`	Data Type	Mandatory ?		Column Name	Data Type	Mandatory	Remarks
				STATUS_DESCRIPTION	VARCHAR2(240)		Do Not Map
				STAT_AMOUNT	NUMBER		Do Not Map
				GROUP_ID	NUMBER(15)		Do Not Map
				REQUEST_ID	NUMBER(15)		Do Not Map
				SUBLEDGER_DOC_SEQUENCE_ID	NUMBER		Do Not Map
				SUBLEDGER_DOC_SEQUENCE_VALUE	NUMBER		Do Not Map

Oracle Retail Invoice Match System to Payables Open Interface Tables

IM_AP_STAGE_HEAD	AP_INVOICES_INTERFACE	COMMENTS
DOC_ID	INVOICE_ID	
SEQ_NO	NONE	
INVOICE_TYPE_LOOKUP_CODE	INVOICE_TYPE_LOOKUP_CODE	
INVOICE_NUMBER	INVOICE_NUM	
VENDOR	~~VENDOR_ID~~ **VENDOR_NUM**	
ORACLE_SITE_ID	VENDOR_SITE_ID	
CURRENCY_CODE	INVOICE_CURRENCY_CODE	
EXCHANGE_RATE	EXCHANGE_RATE	
EXCHANGE_RATE_TYPE	EXCHANGE_RATE_TYPE	Expecting the value of '**USER**' in this column

IM_AP_STAGE_HEAD	AP_INVOICES_INTERFACE	COMMENTS
DOC_DATE	INVOICE_DATE	
AMOUNT	INVOICE_AMOUNT	
BEST_TERMS_DATE	TERMS_DATE	
SEGMENT1	ACCTS_PAY_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to ACCTS_PAY_CODE_CONCATENATED
SEGMENT2	ACCTS_PAY_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to ACCTS_PAY_CODE_CONCATENATED
SEGMENT3	ACCTS_PAY_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to ACCTS_PAY_CODE_CONCATENATED
SEGMENT4	ACCTS_PAY_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to ACCTS_PAY_CODE_CONCATENATED
SEGMENT5	ACCTS_PAY_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to ACCTS_PAY_CODE_CONCATENATED
SEGMENT6	ACCTS_PAY_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to ACCTS_PAY_CODE_CONCATENATED
SEGMENT7	ACCTS_PAY_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to ACCTS_PAY_CODE_CONCATENATED

IM_AP_STAGE_HEAD	AP_INVOICES_INTERFACE	COMMENTS
SEGMENT8	ACCTS_PAY_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to ACCTS_PAY_CODE_CONCATENATED
SEGMENT9	ACCTS_PAY_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to ACCTS_PAY_CODE_CONCATENATED
SEGMENT10	ACCTS_PAY_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to ACCTS_PAY_CODE_CONCATENATED
CREATE_DATE_TIME	CREATION_DATE	

IM_AP_STAGE_DETAIL	AP_INVOICE_LINES_INTERFACE	COMMENTS
DOC_ID	INVOICE_ID	
SEQ_NO	LINE_NUM	
TRAN_CODE	NONE	
LINE_TYPE_LOOKUP_CODE	LINE_TYPE_LOOKUP_CODE	
AMOUNT	AMOUNT	
VAT_CODE	TAX_CODE	
SEGMENT1	DIST_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to DIST_CODE_CONCATENATED
SEGMENT2	DIST_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to DIST_CODE_CONCATENATED
SEGMENT3	DIST_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to DIST_CODE_CONCATENATED
SEGMENT4	DIST_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to DIST_CODE_CONCATENATED
SEGMENT5	DIST_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to DIST_CODE_CONCATENATED
SEGMENT6	DIST_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to DIST_CODE_CONCATENATED

IM_AP_STAGE_DETAIL	AP_INVOICE_LINES_INTERFACE	COMMENTS
SEGMENT7	DIST_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to DIST_CODE_CONCATENATED
SEGMENT8	DIST_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to DIST_CODE_CONCATENATED
SEGMENT9	DIST_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to DIST_CODE_CONCATENATED
SEGMENT10	DIST_CODE_CONCATENATED	All segments (1-10) concatenated with '-' will be mapped to DIST_CODE_CONCATENATED
CREATE_DATE_TIME	CREATION_DATE	

IM_FINANCIALS_STAGE_V	GL_INTERFACE	COMMENTS
STATUS	STATUS	'NEW' hardcoded value of NEW in the view defiinition.
ACTUAL_FLAG	ACTUAL_FLAG	'A' , hardcoded value of 'A' in the view definition
TRAN_CODE	REFERENCE23	
DEBIT_CREDIT_IND		
DOC_ID	REFERENCE22	
PARENT_ID		
DOC_DATE	ACCOUNTING_DATE	
RECEIPT_ID	REFERENCE25	
RECEIPT_DATE		
VENDOR_TYPE		
VENDOR	REFERENCE20	

IM_FINANCIALS_STAGE_V	GL_INTERFACE	COMMENTS
ORDER_NO	REFERENCE24	
CURRENCY_CODE	CURRENCY_CODE	
AMOUNT		
BEST_TERMS		
BEST_TERMS_DATE		
MANUALLY_PAID_IND		
PRE_PAID_IND		
CREATE_ID	CREATED_BY	
CREATE_DATETIME	DATE_CREATED	
SEGMENT1	SEGMENT1	
SEGMENT2	SEGMENT2	
SEGMENT3	SEGMENT3	
SEGMENT4	SEGMENT4	
SEGMENT5	SEGMENT5	
SEGMENT6	SEGMENT6	
SEGMENT7	SEGMENT7	
SEGMENT8	SEGMENT8	
SEGMENT9	SEGMENT9	
SEGMENT10	SEGMENT10	
VAT_CODE	TAX_CODE	
VAT_RATE		
DEAL_ID		
LOCAL_CURRENCY		
INCOME_LOCAL_CURRENCY		
TOTAL_COST_INC_VAT		
EXT_DOC_ID	REFERENCE21	
SET_OF_BOOKS_ID	SET_OF_BOOKS_ID	
USER_JE_SOURCE_NAME	USER_JE_SOURCE_NAME	Will have a constant value "**Retail Invoices**"

IM_FINANCIALS_STAGE_V	GL_INTERFACE	COMMENTS
USER_JE_CATEGORY_NAME	USER_JE_CATEGORY_NAME	Will have one of the following values (“**Writeoffs**”, “**Prepayments**”, “**Manual Payments**”)
ENTERED_DR	ENTERED_DR	
ENTERED_CR	ENTERED_CR	