Interfaces and Conversions

Interfaces and Conversions

Interface/Conversion examples and details:
The below list of interfaces/conversions are covered in this section. Details like pre-requisites required, interface tables, interface program, base tables, validations that need to be performed after inserting the details into the interface tables and required columns that need to be populated in the interface table are discussed for each interface.

P2p cycle

1. Requisitions
2. Purchase orders
3. Receiving
4. AP invoices

O2c cycle

1. order import interface (sales order conversion)
2. AR receipts
3. Auto invoice interface

Inventory

1. Item import (item conversion)
2. Inventory on-hand quantity interface

General ledger:

1. Journal import
2. Budget import
3. Daily conversion rates

Miscellaneous:

1. Customer conversion
2. Vendor conversion
3. Lockbox interface

❖ Requisition import

You can automatically import requisitions into Oracle Applications using the Requisitions Open Interface

1. Pre-requisites:

Set of Books
Code combinations
Employees
Items
Define a Requisition Import Group-By method in the Default region of the Purchasing Options window.
Associate a customer with your deliver-to location using the Customer Addresses window for internally sourced requisitions.

2. Interface tables:

PO_REQUISITIONS_INTERFACE_ALL
PO_REQ_DIST_INTERFACE_ALL

3. Base tables:

PO_REQUISITIONS_HEADERS_ALL
PO_REQUISITION_LINES_ALL
PO_REQ_DISTRIBUTIONS_ALL

4. Concurrent program:

REQUISITION IMPORT

5. Validations:

Check for interface transaction source code, requisition destination type.
Check for quantity ordered, authorization status type.

6. Some important columns that need to be populated in the interface tables:

PO_REQUISITIONS_INTERFACE_ALL:
INTERFACE_SOURCE_CODE (to identify the source of your imported Requisitions)
DESTINATION_TYPE_CODE
AUTHORIZATION_STATUS
PREPARER_ID or PREPARER_NAME
QUANTITY
CHARGE_ACCOUNT_ID or charge account segment values
DESTINATION_ORGANIZATION_ID or DESTINATION_ORGANIZATION_CODE
DELIVER_TO_LOCATION_ID or DELIVER_TO_LOCATION_CODE
DELIVER_TO_REQUESTOR_ID or DELIVER_TO_REQUESTOR_NAME
ORG_ID
ITEM_ID or item segment values (values if the SOURCE_TYPE_CODE or DESTINATION_TYPE_CODE is 'INVENTORY')

PO_REQ_DIST_INTERFACE_ALL:

CHARGE_ACCOUNT_ID or charge account segment values
DISTRIBUTION_NUMBER
DESTINATION_ORGANIZATION_ID
DESTINATION_TYPE_CODE
INTERFACE_SOURCE_CODE
ORG_ID
DIST_SEQUENCE_ID (if MULTI_DISTRIBUTIONS is set to Y)

❖ Purchase Order conversion:

The Purchasing Document Open Interface concurrent program was replaced by two new concurrent programs Import Price Catalogs and Import Standard Purchase Orders. Import Price Catalogs concurrent program is used to import Catalog Quotations, Standard Quotations, and Blanket Purchase Agreements. Import Standard Purchase Orders concurrent program is used to import Unapproved or Approved Standard Purchase Orders.

Import Standard Purchase Orders

1. Pre-requisites:

Suppliers, sites and contacts
Buyers
Line Types
Items
PO Charge account setup

2. Interface Tables:

PO_HEADERS_INTERFACE
PO_LINES_INTERFACE
PO_DISTRIBUTIONS_INTERFACE
PO_INTERFACE_ERRORS (Fallouts)

3. Interface Program:

Import Standard Purchase Orders.

4. Base Tables:

PO_HEADERS_ALL
PO_LINES_ALL
PO_DISTRIBUTIONS_ALL
PO_LINE_LOCATIONS_ALL

5. Validations:

Header:
Check if OU name is valid
Check if Supplier is valid
Check if Supplier site is valid
Check if buyer is valid
Check if Payment term is valid
Check if Bill to and ship to are valid
Check if FOB, freight terms are valid

Lines:

Check if Line_type, ship_to_org, item, uom, ship_to_location_id, requestor, charge_account, deliver_to_location are valid

General:
Check for duplicate records in interface tables
Check if the record already exists in base tables.

6. Some important columns that need to be populated in the interface tables:

PO_HEADERS_INTERFACE:
INTERFACE_HEADER_ID (PO_HEADERS_INTERFACE_S.NEXTVAL), BATCH_ID, ORG_ID, INTERFACE_SOURCE_CODE, ACTION ('ORIGINAL','UPDATE','REPLACE'), GROUP_CODE, DOCUMENT_TYPE_CODE, PO_HEADER_ID (NULL), RELEASE_ID, RELEASE_NUM, CURRENCY_CODE, RATE, AGENT_NAME, VENDOR_ID, VENDOR_SITE_ID, SHIP_TO_LOCATION, BILL_TO_LOCATION, PAYMENT_TERMS

PO_LINES_INTERFACE:
INTERFACE_LINE_ID, INTERFACE_HEADER_ID, LINE_NUM, SHIPMENT_NUM, ITEM, REQUISITION_LINE_ID, UOM, UNIT_PRICE, FREIGHT_TERMS, FOB

PO_DISTRIBUTIONS_INTERFACE:
INTERFACE_LINE_ID, INTERFACE_HEADER_ID, INTERFACE_DISTRIBUTION_ID, DISTRIBUTION_NUM, QUANTITY_ORDERED, QTY_DELIVERED, QTY_BILLED, QTY_CANCELLED, DELIVER_TO_LOCATION_ID, DELIVER_TO_PERSON_ID, SET_OF_BOOKS, CHARGE_ACCT, AMOUNT_BILLED.

Import Blanket Purchase Agreements:

1. Interface Tables:
   PO_HEADERS_INTERFACE
   PO_LINES_INTERFACE

2. Interface program:
   Import Price Catalogs

3. Base tables:
   PO_HEADERS_ALL
   PO_LINES_ALL
   PO_LINE_LOCATIONS_ALL

Example:
Suppose you want to create a blanket with one line and two price breaks and the details for the price break are as below:
1) Quantity = 500, price = 10, effective date from '01-JAN-2006′ to '31-JUN-2006′
2) Quantity = 500, price = 11, effective date from '01-JUL-2006′ to '01-JAN-2007′

To create the above the BPA, you would create ONE record in PO_HEADERS_INTERFACE and THREE records in PO_LINES_INTERFACE
LINE1: It will have only the line information. LINE NUM would be 1.

LINE2: For the first Price Break details, LINE NUM will be the same as above i.e. 1. SHIPMENT_NUM would be 1 and SHIPMENT_TYPE would be 'PRICE BREAK'

LINE3: For the second Price Break details, LINE NUM will be the same as above i.e. 1. SHIPMENT_NUM would be 2 and SHIPMENT_TYPE would be 'PRICE BREAK'

All the line-level records above must have the same INTERFACE_HEADER_ID.
For detailed explanation refer to the below article:
http://www.erpschools.com/Apps/oracle-applications/articles/financials/Purchasing/Import-Blanket-Purchase-Agreements/index.aspx

❖ PO Receipts Interface

The Receiving Open Interface is used for processing and validating receipt data that comes from sources other than the Receipts window in Purchasing.

1. Pre-requisites:

Set of Books
Code combinations
Employees
Items

2. Interface tables:

RCV_HEADERS_INTERFACE
RCV_TRANSACTIONS_INTERFACE
PO_INTERFACE_ERRORS

3. Concurrent program:

RECEIVING OPEN INTERFACE

4. Base tables:

RCV_SHIPMENT_HEADERS
RCV_SHIPMENT_LINES
RCV_TRANSACTIONS

5. Validations:

Check that SHIPPED_DATE should not be later than today.
Check if vendor is valid.
If Invoice number is passed, check for its validity
Check if Item is valid

6. Some important columns that need to be populated in the interface tables:

RCV_HEADERS_INTERFACE:
HEADER_INTERFACE_ID
GROUP_ID
PROCESSING_STATUS_

CODE
RECEIPT_SOURCE_CODE
TRANSACTION_TYPE
SHIPMENT_NUM
RECEIPT_NUM
VENDOR_NAME
SHIP_TO_
ORGANIZATION_CODE
SHIPPED_DATE
INVOICE_NUM
INVOICE_DATE
TOTAL_INVOICE_
AMOUNT
PAYMENT_TERMS_ID
EMPLOYEE_NAME
VALIDATION_FLAG (Indicates whether to validate a row or not, values 'Y', 'N')

RCV_TRANSACTIONS_INTERFACE:
INTERFACE_TRANSACTION_ID
GROUP_ID
TRANSACTION_TYPE ('SHIP' for a standard shipment (an ASN or ASBN)
or 'RECEIVE' for a standard receipt)
TRANSACTION_DATE
PROCESSING_STATUS_CODE ='PENDING'
CATEGORY_ID
QUANTITY
UNIT_OF_MEASURE
ITEM_DESCRIPTION
ITEM_REVISION
EMPLOYEE_ID
AUTO_TRANSACT_CODE
SHIP_TO_LOCATION_ID
RECEIPT_SOURCE_CODE
TO_ORGANIZATION_CODE
SOURCE_DOCUMENT_CODE
PO_HEADER_ID
PO_RELEASE_ID
PO_LINE_ID
PO_LINE_LOCATION_ID
PO_DISTRIBUTION_ID
SUBINVENTORY
HEADER_INTERFACE_ID
DELIVER_TO_PERSON_NAME
DELIVER_TO_LOCATION_CODE
VALIDATION_FLAG
ITEM_NUM
VENDOR_ITEM_NUM
VENDOR_ID
VENDOR_SITE_ID

ITEM_ID
ITEM_DESCRIPTION
SHIP_TO_LOCATION_ID

❖ AP invoice interface

This interface helps us to import vendor invoices into Oracle applications from external systems into Oracle Applications.

1. Pre-requisites:
   Set of Books
   Code combinations
   Employees
   Lookups

2. Interface tables:
   AP_INVOICES_INTERFACE
   AP_INVOICE_LINES_INTERFACE

3. Base tables:
   AP_INVOICES_ALL – header information
   AP_INVOICE_DISTRIBUTIONS_ALL – lines info

4. Concurrent program:
   Payables Open Interface Import

5. Validations:
   Check for valid vendor
   Check for Source, Location, org_id, currency_code's validity
   Check for valid vendor site code.
   Check if record already exists in payables interface table.

6. Some important columns that need to be populated in the interface tables:

AP_INVOICES_INTERFACE:

INVOICE_ID
INVOICE_NUM
INVOICE_DATE
VENDOR_NUM
VENDOR_SITE_ID
INVOICE_AMOUNT
INVOICE_CURRENCY_CODE
EXCHANGE_RATE
EXCHANGE_RATE_TYPE
EXCHANGE_DATE
DESCRIPTION
SOURCE
PO_NUMBER
PAYMENT_METHOD_LOOKUP_CODE
PAY_GROUP_LOOKUP_CODE
ATTRIBUTE1 TO 15
ORG_ID

AP_INVOICE_LINES_INTERFACE:

INVOICE_ID
INVOICE_LINE_ID
LINE_TYPE_LOOKUP_CODE
AMOUNT
DESCRIPTION
TAX_CODE
PO_NUMBER
PO_LINE_NUMBER
PO_SHIPMENT_NUM
PO_DISTRIBUTION_NUM
PO_UNIT_OF_MEASURE
QUANTITY_INVOICED
DIST_CODE_CONCATENATED
DIST_CODE_COMBINATION_ID
ATTRIBUTE1
ATTRIBUTE2
ATTRIBUTE3
ATTRIBUTE4
ATTRIBUTE5
ORG_ID

❖ Order Import Interface (Sales Order Conversion)

Order Import enables you to import Sales Orders into Oracle Applications instead of manually entering them.

1. Pre-requisites:
   Order Type
   Line Type
   Items
   Customers
   Ship Method/ Freight Carrier
   Sales Person
   Sales Territories
   Customer Order Holds
   Sub Inventory/ Locations
   On hand Quantity

2. Interface tables:
   - OE_HEADERS_IFACE_ALL
   - OE_LINES_IFACE_ALL
   - OE_ACTIONS_IFACE_ALL
   - OE_ORDER_CUST_IFACE_ALL
   - OE_PRICE_ADJS_IFACE_ALL
   - OE_PRICE_ATTS_IFACE_ALL

3. Base tables:

- OE_ORDER_HEADERS_ALL
- OE_ORDER_LINES_ALL

Pricing tables: QP_PRICING_ATTRIBUTES

4. Concurrent Program:

Order Import

5. Validations:

- Check for sold_to_org_id. If does not exist, create new customer by calling create_new_cust_info API.
- Check for sales_rep_id. Should exist for a booked order.
- Ordered_date should exist (header level)
- Delivery_lead_time should exist (line level)
- Earliest_acceptable_date should exist.
- Freight_terms should exist

6. Notes:

During import of orders, shipping tables are not populated.

1. If importing customers together with the order, OE_ORDER_CUST_IFACE_ALL has to be populated and the base tables are HZ_PARTIES, HZ_LOCATIONS.

Orders can be categorized based on their status:

- Entered orders
- Booked orders
- Closed orders

Order Import API OE_ORDER_PUB.GET_ORDER and PROCESS_ORDER can also be used to import orders.

7.Some important columns that need to populated in the interface tables:

OE_HEADERS_IFACE_ALL:

- ORIG_SYS_DOCUMENT_REF
- ORDER_SOURCE
- CONVERSION_RATE
- ORG_ID
- ORDER_TYPE_ID
- PRICE_LIST
- SOLD_FROM_ORG_ID
- SOLD_TO_ORG_ID
- SHIP_TO_ORG_ID
- SHIP_FROM_ORG_ID
- CUSTOMER_NAME
- INVOICE_TO_ORG_ID
- OPERATION_CODE

OE_LINES_IFACE_ALL:

- ORDER_SOURCE_ID
- ORIG_SYS_DOCUMENT_REF
- ORIG_SYS_LINE_REF
- ORIG_SYS_SHIPMENT_REF

- INVENTORY_ITEM_ID
- LINK_TO_LINE_REF
- REQUEST_DATE
- DELIVERY_LEAD_TIME
- DELIVERY_ID
- ORDERED_QUANTITY
- ORDER_QUANTITY_UOM
- SHIPPING_QUANTITY
- PRICING_QUANTITY
- PRICING_QUANTITY_UOM
- SOLD_FROM_ORG_ID
- SOLD_TO_ORG_ID
- INVOICE_TO_ ORG_ID
- SHIP_TO_ORG_ID
- PRICE_LIST_ID
- PAYMENT_TERM_ID

❖ Receipt API

To bring in Unapplied Receipts and Conversion Receipts for Open Debit items to reduce the balance to the original amount due.

1. Pre-requisites:

Set of Books
Code combinations
Items
Quick Codes
Sales representatives
Customers
Sales Tax rate

2. API:

AR_RECEIPT_API_PUB.CREATE_CASH
AR_RECEIPT_API_PUB.CREATE_AND_APPLY

3. Base tables:

AR_CASH_RECEIPTS

4. Validations:

Check the currency and the exchange rate type to assign the exchange rate.
Validate bill to the customer.
Get bill to site use id.
Get the customer trx id for this particular transaction number.
Get payment schedule date for the customer trx id.

❖ Auto Invoice interface

This interface is used to import Customer invoices, Credit memos, Debit memos and On Account credits.

1. Pre-requisites:

Set of Books
Code combinations
Items
Sales representatives
Customers
Sales Tax rate
Payment Terms
Transaction Types
Freight Carriers
FOB
Batch Sources
Accounting Rules

2. Interface tables:

RA_INTERFACE_LINES_ALL
RA_INTERFACE_SALESCREDITS
RA_INTERFACE_DISTRIBUTIONS
RA_INTERFACE_ERRORS (details about the failed records)

3. Base tables:

RA_BATCHES
RA_CUSTOMER_TRX_ALL
RA_CUSTOMER_TRX_LINES_ALL
AR_PAYMENT_SCHEDULES_ALL RA_CUSTOMER_TRX_LINE_SALESREPS
RA_CUST_TRX_GL_DIST_ALL
RA_CUSTOMER_TRX_TYPES_ALL

4. Concurrent Program:

Auto invoice master program

5. Validations:

Check for amount, batch source name, conversion rate, conversion type.
Validate orig_system_bill_customer_id, orig_system_bill_address_id, quantity.
Validate if the amount includes tax flag.

6. Some important columns that need to be populated in the interface tables:

RA_INTERFACE_LINES_ALL:
AGREEMENT_ID
COMMENTS
CONVERSION_DATE
CONVERSION_RATE
CONVERSION_TYPE
CREDIT_METHOD_FOR_ACCT_RULE
CREDIT_METHOD_FOR_INSTALLMENTS
CURRENCY_CODE
CUSTOMER_BANK_ACCOUNT_ID
CUST_TRX_TYPE_ID
DOCUMENT_NUMBER
DOCUMENT_NUMBER_SEQUENCE_ID
GL_DATE
HEADER_ATTRIBUTE1–15
HEADER_ATTRIBUTE_CATEGORY
INITIAL_CUSTOMER_TRX_ID
INTERNAL_NOTES
INVOICING_RULE_ID
ORIG_SYSTEM_BILL_ADDRESS_ID
ORIG_SYSTEM_BILL_CONTACT_ID
ORIG_SYSTEM_BILL_CUSTOMER_ID
ORIG_SYSTEM_SHIP_ADDRESS_ID
ORIG_SYSTEM_SHIP_CONTACT_ID
ORIG_SYSTEM_SHIP_CUSTOMER_ID

ORIG_SYSTEM_SOLD_CUSTOMER_ID
ORIG_SYSTEM_BATCH_NAME
PAYMENT_SERVER_ORDER_ID
PREVIOUS_CUSTOMER_TRX_ID
PRIMARY_SALESREP_ID
PRINTING_OPTION
PURCHASE_ORDER
PURCHASE_ORDER_DATE
PURCHASE_ORDER_REVISION
REASON_CODE
RECEIPT_METHOD_ID
RELATED_CUSTOMER_TRX_ID
SET_OF_BOOKS_ID
TERM_ID
TERRITORY_ID
TRX_DATE
TRX_NUMBER

❖ Item import (Item conversion):

The Item Interface lets you import items into Oracle Inventory.

1. Pre-requisites:

- Creating an Organization
- Code Combinations
- Templates
- Defining Item Status Codes
- Defining Item Types

2. Interface tables:

- MTL_SYSTEM_ITEMS_INTERFACE
- MTL_ITEM_REVISIONS_INTERFACE (If importing revisions)
- MTL_ITEM_CATEGORIES_INTERFACE (If importing categories)
- MTL_INTERFACE_ERRORS (View errors after import)

3. Concurrent Program:

Item import

In the item import parameters form, for the parameter 'set process id', specify the 'set process id' value given in the mtl_item_categories_interface table. The parameter 'Create or Update' can have any value. Through the import process, we can only create item category assignment(s). Updating or Deletion of item category assignment is not supported.

4. Base tables:

- MTL_SYSTEM_ITEMS_B
- MTL_ITEM_REVISIONS_B
- MTL_CATEGORIES_B
- MTL_CATEGORY_SETS_B
- MTL_ITEM_STATUS
- MTL_ITEM_TEMPLATES

5. Validations:

- Check for valid item type.
- Check for valid part_id/segment of the source table.
- Validate part_id/segment1 for master org.
- Validate and translate template id of the source table.
- Check for valid template id. (Attributes are already set for items, default attributes for
- that template, i.e., purchasable, stockable, etc)
- Check for valid item status.
- Validate primary uom of the source table.
- Validate attribute values.
- Validate other UOMs of the source table.
- Check for unique item type. Discard the item, if part has non-unique item type.
- Check for description, inv_um uniqueness
- Validate organization id.
- Load master records and category records only if all validations are passed.
- Load child record if no error found.

6. Some important columns that need to populated in the interface tables:

MTL_SYSTEM_ITEMS_INTERFACE:

PROCESS_FLAG = 1 (1= Pending, 2= Assign Complete,
3= Assign/Validation Failed, 4= Validation succeeded; Import failed, 5 = Import in Process,
7 = Import succeeded)

TRANSACTION_TYPE = 'CREATE', 'UPDATE'

SET_PROCESS_ID = 1

ORGANIZATION_ID
DESCRIPTION
ITEM_NUMBER and/or SEGMENT (n)
MATERIAL_COST
REVISION
TEMPLATE_ID
SUMMARY_FLAG
ENABLED_FLAG
PURCHASING_ITEM_FLAG
SALES_ACCOUNT (defaulted from]
MTL_PARAMETERS.SALES_ACCOUNT)
COST_OF_SALES_ACCOUNT (defaulted from MTL_PARAMETERS.
COST_OF_SALES_ACCOUNT)

MTL_ITEM_CATEGORIES_INTERFACE:

INVENTORY_ITEM_ID or ITEM_NUMBER.
ORGANIZATION_ID or ORGANIZATION_CODE or both.
TRANSACTION_TYPE = 'CREATE' ('UPDATE' or 'DELETE' is not possible through Item Import).
CATEGORY_SET_ID or CATEGORY_SET_NAME or both.
CATEGORY_ID or CATEGORY_NAME or both.
PROCESS_FLAG = 1
SET_PROCESS_ID (The item and category interface records should have the same set_process_id, if you are importing item and category assignment together)

MTL_ITEM_REVISIONS_INTERFACE:

- INVENTORY_ITEM_ID or ITEM_NUMBER (Must match the item_number in mtl_system_items_interface table)
- ORGANIZATION_ID or ORGANIZATION_CODE or both
- REVISION
- CHANGE_NOTICE
- ECN_INITIATION_DATE
- IMPLEMENTATION_DATE
- IMPLEMENTED_SERIAL_NUMBER
- EFFECTIVITY_DATE
- ATTRIBUTE_CATEGORY
- ATTRIBUTEn
- REVISED_ITEM_SEQUENCE_ID
- DESCRIPTION
    - PROCESS_FLAG = 1
    - TRANSACTION_TYPE = 'CREATE'
    - SET_PROCESS_ID = 1
- Each row in the mtl_item_revisions_interface table must have the REVISION
- and EFFECTIVITY_DATE in alphabetical (ASCII sort) and chronological order.

❖ Inventory On-hand quantity Interface

This interface lets you import the on hand inventory into Oracle.

1. Interface tables:

MTL_TRANSACTIONS_INTERFACE

MTL_MTL_TRANSACTION_LOTS_INTERFACE (If the item is Lot controlled)
MTL_SERIAL_NUMBERS_INTERFACE (If the item is Serial controlled)

2. Concurrent Program:

Launch the **Transaction Manager** through Interface Manager or explicitly call the API – INV_TXN_MANAGER_PUB.PROCESS_TRANSACTIONS () to launch a dedicated transaction worker to process them.

The Transaction Manager picks up the rows to process based on the LOCK_FLAG, TRANSACTION_MODE, and PROCESS_FLAG. Only records with TRANSACTION_MODE of 3, LOCK_FLAG of '2′, and PROCESS_FLAG of '1′ will be picked up by the Transaction Manager and assigned to a Transaction Worker. If a record fails to process completely, then PROCESS_FLAG will be set to '3′ and ERROR_CODE and ERROR_EXPLANATION will be populated with the cause for the error.

3. Base Tables:

MTL_ON_HAND_QUANTITIES
MTL_LOT_NUMBERS
MTL_SERIAL_NUMBERS

4. Validations:

Validate organization_id
Check if item is assigned to organization
Validate disposition_id
Check if the item for the org is lot controlled before inserting into the Lots interface table.
Check if the item for the org is serial controlled before inserting into Serial interface table.
Check if inventory already exists for that item in that org and for a lot.
Validate organization_id, organization_code.
Validate inventory item id.
Transaction period must be open.

5. Some important columns that need to be populated in the interface tables:

MTL_TRANSACTIONS_INTERFACE:

TRANSACTION_SOURCE_NAME (ANY USER DEFINED VALUE),
TRANSACTION_HEADER_ID (MTL_MATERIAL_TRANSACTIONS_S.NEXTVAL)
TRANSACTION_INTERFACE_ID (MTL_MATERIAL_TRANSACTIONS_S.NEXTVAL – If item is lot or serial controlled, use this field to link to mtl_transactions_interface otherwise leave it as NULL),
TRANSACTION_DATE,
TRANSACTION_TYPE_ID,
PROCESS_FLAG (1 = Yet to be processed, 2 = Processed, 3= Error)
TRANSACTION_MODE (2 = Concurrent – to launch a dedicated transaction worker
to explicitly process a set of transactions.
3 = Background – will be picked up by transaction manager
polling process and assigned to transaction
worker. These will not be picked up until the
transaction manager is running)
SOURCE_CODE,
SOURCE_HEADER_ID,
SOURCE_LINE_ID (Details about the source like Order Entry etc for tracking purposes)
TRANSACTION_SOURCE_ID

Source Type	Foreign Key Reference
Account	GL_CODE_COMBINATIONS.CODE_COMBINATION_ID
Account Alias	MTL_GENERIC_DISPOSITIONS.DISPOSITION_ID
Job or schedule	WIP_ENTITIES.WIP_ENTITY_ID
Sales Order	MTL_SALES_ORDERS.SALES_ORDER_ID

ITEM_SEGMENT1 TO 20,
TRANSACTION_QTY,
TRANSACTION_UOM,
SUBINVENTORY_CODE,
ORGANIZATION_ID,
LOC_SEGMENT1 TO 20.

MTL_TRANSACTION_LOTS_INTERFACE:
TRANSACTION_INTERFACE_ID,
LOT_NUMBER,
LOT_EXPIRATION_DATE,
TRANSACTION_QUANTITY,
SERIAL_TRANSACTION_TEMP_ID (This is required for items under both lot and serial control to identify child records in mtl_serial_numbers_interface)

MTL_SERIAL_NUMBERS_INTERFACE:
TRANSACTION_INTERFACE_ID,
FM_SERIAL_NUMBER,
TO_SERIAL_NUMBER,
VENDOR_SERIAL_NUMBER

❖ GL Journal interface

This interface lets you import journals from other applications like Receivables, Payables etc to integrate the information with General Ledger.

1. Pre-requisites:

Set of Books
Flex field Value sets
Code Combinations
Currencies
Categories
Journal Sources

2. Interface tables:

GL_INTERFACE

3. Base tables:

GL_JE_HEADERS
GL_JE_LINES
GL_JE_BACTHES

4. Concurrent Program:

Journal Import
Journal Posting — populates GL_BALANCES

5. Validations:

Validate SOB, journal source name, journal category name, actual flag
A – Actual amounts
B – Budget amounts
E – Encumbrance amount
If you enter E in the interface table, then enter appropriate encumbrance ID, if
B enter budget id.
Check if accounting date or GL date based period name is valid (i.e., not closed).
Check if accounting date falls in open or future open period status.
Check chart of accounts id based on Sob id.
Check if code combination is valid and enabled.
Check if record already exists in GL interface table.
Check if already journal exists in GL application.

6. Some important columns that need to be populated in the interface tables:

GL_INTERFACE:
STATUS
SET_OF_BOOKS_ID
ACCOUNTING_DATE
CURRENCY_CODE
DATE_CREATED
CREATED_BY

ACTUAL_FLAG
USER_JE_CATEGORY_NAME
USER_JE_SOURCE_NAME
CURRENCY_CONVERSION_DATE
ENCUMBRANCE_TYPE_ID
BUDGET_VERSION_ID
USER_CURRENCY_CONVERSION_TYPE
CURRENCY_CONVERSION_RATE
SEGMENT1 to
ENTERED_DR
ENTERED_CR
ACCOUNTED_DR
ACCOUNTED_CR
TRANSACTION_DATE
PERIOD_NAME
JE_LINE_NUM
CHART_OF_ACCOUNTS_ID
FUNCTIONAL_CURRENCY_CODE
CODE_COMBINATION_ID
DATE_CREATED_IN_GL
GROUP_ID

❖ GL budget interface

Budget interface lets you load budget data from external sources into Oracle Applications.

1. Pre-requisites:
   Set of Books
   Flex field Value sets
   Code Combinations

2. Interface tables:
   GL_BUDGET_INTERFACE

3. Base tables:
   GL_BUDGETS
   GL_BUDGET_ASSIGNMENTS
   GL_BUDGET_TYPES

4. Concurrent program:
   Budget Upload

5. Validations:
   Check if CURRENCY_CODE is valid.
   Check if SET_OF_BOOKS_ID is valid.
   Check if BUDGET_ENTITY_NAME (budget organization) is valid.

6. Some important columns that need to be populated in the interface tables:

GL_BUDGET_INTERFACE:

BUDGET_NAME NOT
BUDGET_ENTITY_NAME
CURRENCY_CODE
FISCAL_YEAR
UPDATE_LOGIC_TYPE
BUDGET_ENTITY_ID
SET_OF_BOOKS_ID
CODE_COMBINATION_ID
BUDGET_VERSION_ID
PERIOD_TYPE
DR_FLAG
STATUS
ACCOUNT_TYPE
PERIOD1_AMOUNT through PERIOD60_AMOUNT
SEGMENT1 through SEGMENT30

❖ GL daily conversion rates

This interface lets you load the rates automatically into General Ledger.

1. Pre-requisites:
   Currencies
   Conversion rate Types

2. Interface tables:
   GL_DAILY_RATES_INTERFACE

3. Base tables:
   GL_DAILY_RATES
   GL_DAILY_CONVERSION_TYPES

4. Concurrent Program:
   You do not need to run any import programs. The insert, update, or deletion of rates in GL_DAILY_RATES is done automatically by database triggers on the GL_DAILY_RATES_INTERFACE. All that is required is to develop program to populate the interface table with daily rates information.

5. Validations:
   Check if FROM_CURRENCY and TO_CURRENCY are valid.
   Check if USER_CONVERSION_TYPE is valid.

6.Some important columns that need to be populated in the interface tables:

GL_DAILY_RATES_INTERFACE:

FROM_CURRENCY
TO_CURRENCY
FROM_CONVERSION_DATE

TO_CONVERSION_DATE
USER_CONVERSION_TYPE
CONVERSION_RATE
MODE_FLAG (D= Delete, I = Insert, U = Update)
INVERSE_CONVERSION_RATE

❖ Customer conversion

Customer Interface helps you create customers in Oracle Applications.

1. Interface tables:

RA_CUSTOMERS_INTERFACE_ALL
RA_CUSTOMER_PROFILES_INT_ALL
RA_CONTACT_PHONES_INT_ALL
RA_CUSTOMER_BANKS_INT_ALL
RA_CUST_PAY_METHOD_INT_ALL

2. Base tables:

RA_CUSTOMERS
RA_ADDRESSES_ALL
RA_CUSTOMER_RELATIONSHIPS_ALL
RA_SITE_USES_ALL

3. Concurrent program:

Customer Interface

4. Validations:

Check if legacy values fetched are valid.
Check if customer address site is already created.
Check if customer site use is already created.
Check is customer header is already created.
Check whether the ship_to_site has associated bill_to_site
Check whether associated bill_to_site is created or not.
Profile amounts validation:
Validate cust_account_id, validate customer status.
Check if the location already exists in HZ_LOCATIONS. If does not exist, create new location.

5. Some important columns that need to be populated in the interface tables:

RA_CUSTOMERS_INTERFACE_ALL:
ORIG_SYSTEM_CUSTOMER_REF
SITE_USE_CODE
ORIG_SYSTEM_ADDRESS_REF
INSERT_UPDATE_FLAG (I = Insert, U = Update)
CUSTOMER_NAME
CUSTOMER_NUMBER
CUSTOMER_STATUS
PRIMARY_SITE_USE_FLAG

LOCATION
ADDRESS1
ADDRESS2
ADDRESS3
ADDRESS4
CITY
STATE
PROVINCE
COUNTY
POSTAL_CODE
COUNTRY
CUSTOMER_ATTRIBUTE1
CUSTOMER_ATTRIBUTE2
CUSTOMER_ATTRIBUTE3
CUSTOMER_ATTRIBUTE4
CUSTOMER_ATTRIBUTE5
LAST_UPDATED_BY
LAST_UPDATE_DATE
CREATED_BY
CREATION_DATE
ORG_ID
CUSTOMER_NAME_PHONETIC

RA_CUSTOMER_PROFILES_INT_ALL:
INSERT_UPDATE_FLAG
ORIG_SYSTEM_CUSTOMER_REF
ORIG_SYSTEM_ADDRESS_REF
CUSTOMER_PROFILE_CLASS_NAME
CREDIT_HOLD
LAST_UPDATED_BY
LAST_UPDATE_DATE
CREATION_DATE
CREATED_BY
ORG_ID

RA_CONTACT_PHONES_INT_ALL:
ORIG_SYSTEM_CONTACT_REF
ORIG_SYSTEM_TELEPHONE_REF
ORIG_SYSTEM_CUSTOMER_REF
ORIG_SYSTEM_ADDRESS_REF
INSERT_UPDATE_FLAG
CONTACT_FIRST_NAME
CONTACT_LAST_NAME
CONTACT_TITLE
CONTACT_JOB_TITLE
TELEPHONE
TELEPHONE_EXTENSION
TELEPHONE_TYPE
TELEPHONE_AREA_CODE

LAST_UPDATE_DATE
LAST_UPDATED_BY
LAST_UPDATE_LOGIN
CREATION_DATE
CREATED_BY
EMAIL_ADDRESS
ORG_ID

- **Customer API**

Trading Community Architecture (TCA) is an architecture concept designed to support complex trading communities. These APIs utilize the new TCA model, inserting directly to the HZ tables.

API Details:

1. Set the organization id
   Exec dbms_application_info.set_client_info('204');
2. Create a party and an account
   HZ_CUST_ACCOUNT_V2PUB.CREATE_CUST_ACCOUNT()
   HZ_CUST_ACCOUNT_V2PUB.CUST_ACCOUNT_REC_TYPE
   HZ_PARTY_V2PUB.ORGANIZATION_REC_TYPE
   HZ_CUSTOMER_PROFILE_V2PUB.CUSTOMER_PROFILE_REC_TYPE

3. Create a physical location
   HZ_LOCATION_V2PUB.CREATE_LOCATION()
   HZ_LOCATION_V2PUB.LOCATION_REC_TYPE

4. Create a party site using party_id you get from step 2 and location_id from step 3.
   HZ_PARTY_SITE_V2PUB.CREATE_PARTY_SITE()
   HZ_PARTY_SITE_V2PUB.PARTY_SITE_REC_TYPE

5. Create an account site using account_id you get from step 2 and party_site_id from step 4.
   HZ_CUST_ACCOUNT_SITE_V2PUB.CREATE_CUST_ACCT_SITE()
   HZ_CUST_ACCOUNT_SITE_V2PUB.CUST_ACCT_SITE_REC_TYPE

6. Create an account site use using cust_acct_site_id you get from step 5 ans site_use_code = 'BILL_TO'.
   HZ_CUST_ACCOUNT_SITE_V2PUB.CREATE_CUST_SITE_USE()
   HZ_CUST_ACCOUNT_SITE_V2PUB.CUST_SITE_USE_REC_TYPE
   HZ_CUSTOMER_PROFILE_V2PUB.CUSTOMER_PROFILE_REC_TYPE

6. Base table:

HZ_PARTIES
HZ_PARTY_SITES
HZ_LOCATIONS
HZ_CUST_ACCOUNTS
HZ_CUST_SITE_USES_ALL
HZ_CUST_ACCT_SITES_ALL
HZ_PARTY_SITE_USES

7. Validations:

Check if legacy values fetched are valid.
Check if customer address site is already created.

Check if customer site use is already created.
Check is customer header is already created.
Check whether the ship_to_site has associated bill_to_site
Check whether associated bill_to_site is created or not.
Profile amounts validation:
Validate cust_account_id, validate customer status.
Check if the location already exists in HZ_LOCATIONS. If does not exist, create new location.

For detailed explanation refer to the below article:
http://www.erpschools.com/Apps/oracle-applications/articles/financials/Receivables/Customer-TCA-Architecture-and-API/index.aspx

❖ Lockbox interface

AutoLockbox lets us automatically process receipts that are sent directly to the bank instead of manually feeding them in Oracle Receivables.

AutoLockbox is a three step process:

1. ***Import:*** During this step, Lockbox reads and formats the data from your bank file into interface table AR_PAYMENTS_INTERFACE_ALL using a SQL *Loader script.
2. ***Validation:*** The validation program checks data in this interface table for compatibility with Receivables. Once validated, the data is transferred into QuickCash tables (AR_INTERIM_CASH_RECEIPTS_ALL and AR_INTERIM_CASH_RCPT_LINES_ALL).
3. ***Post QuickCash:*** This step applies the receipts and updates your customer's balances.

1. Pre-Requisites:

Banks
Receipt Class
Payment Method
Receipt Source
Lockbox
Transmission format
AutoCash Rule sets

2. Interface tables:

AR_PAYMENTS_INTERFACE_ALL(Import data from bank file)
AR_INTERIM_CASH_RECEIPTS_ALL
AR_INTERIM_CASH_RCPT_LINES_ALL (Validate data in interface table and place in quick cash tables)

3. Base Tables:

AR_CASH_RECEIPTS
AR_RECEIVABLES_APPLICATIONS
AR_ADJUSTMENTS
AR_DISTRIBUTIONS_ALL
AR_PAYMENT_SCHEDULES_ALL

4. Concurrent program:

Lockbox

5. Validations:

Check for valid record type, transmission record id.
Validate sum of the payments within the transmission.
Identify the lockbox number (no given by a bank to identify a lockbox).

6. Some important columns that need to be populated in the interface tables:

AR_PAYMENTS_INTERFACE_ALL:
STATUS
RECORD_TYPE
LOCKBOX_NUMBER
BATCH_NAME
TRANSIT_ROUTING_NUMBER
ACCOUNT
CHECK_NUMBER
REMITTANCE_AMOUNT
DEPOSIT_DATE
ITEM_NUMBER
CURRENCY_CODE
DEPOSIT_TIME

❖ Vendor conversion/interface

This interface is used to import suppliers, supplier sites and site contacts into Oracle applications.

1. Pre-requisites setup's required:

Payment terms
Pay Groups
CCID
Supplier classifications
Bank Accounts
Employees (if employees have to set up as vendors)

2. Interface tables:

AP_SUPPLIERS_INT
AP_SUPPLIER_SITES_INT
AP_SUP_SITE_CONTACT_INT

3. Base Tables:

PO_VENDORS
PO_VENDOR_SITES_ALL
PO_VENDOR_CONTACTS

4. Interface programs:

Supplier Open Interface Import
Supplier Sites Open Interface Import
Supplier Site Contacts Open Interface Import

5. Validations:

Check if vendor already exists

Check if vendor site already exists
Check if site contact already exists
Check if term is defined.

6. Some important columns that need to be populated in the interface tables:

AP_SUPPLIERS_INT:
VENDOR_NUMBER, VENDOR_NAME, VENDOR_TYPE, STATE_REPORTABLE, FED_REPORTABLE, NUM_1099, TYPE_1099, PAY_GROUP_LOOKUP_CODE, VENDOR_ID is auto generated.

AP_SUPPLIER_SITES_INT:
VENDOR_SITE_ID, ORG_ID, VENDOR_SITE_CODE, INACTIVE_DATE, PAY_SITE, PURCHASING_SITE, SITE_PAYMENT_TERM, ADDRESS1, ADDRESS2.ADDRESS3, CITY, STATE, COUNTRY, ZIP, PH_NUM, FAX_NUMBER, TAX_REPORTING_SITE_FLAG.

AP_SUP_SITE_CONTACTS_INT:
VENDOR_ID, VENDOR_SITE_ID, FIRST_NAME, LAST_NAME, AREA_CODE, PHONE, EMAIL, ORG_ID