webMethods Oracle Applications Adapter

Sample Transaction Definitions User's Guide

VERSION 6.0

webMethods, Inc.
3930 Pender Drive
Fairfax, VA 22030
USA
703.460.2500
http://www.webmethods.com

Document ID: ADAPTER-ORACLEAPPS-STDUG-20040716

Contents

About This Guide

This guide describes the sample transaction definitions available for use by application developers when creating their own adapter services for the Oracle Applications Adapter. It also provides an overview of adapter services and transaction processing, and describes the built-in services provided by the Oracle Applications Adapter. It does not describe how to modify the sample transaction definitions. To learn how to use the adapter to modify sample transaction definitions, see the *webMethods Oracle Applications Adapter User's Guide*.

To use this guide effectively, you should:

- Understand the basic concepts described in the *webMethods Integration Server Administrator's Guide* and the *webMethods Developer User's Guide*.
- Be familiar with the operation of the desired Oracle Applications version and the definition and use of Oracle Applications business transactions.

You can also create your own transaction definitions based on the sample transaction definitions. For more information, see the *webMethods Oracle Applications Adapter User's Guide*.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Italic	Identifies variable information that you must supply or change based on your specific situation or environment. Identifies terms the first time they are defined in text. Also identifies service input and output variables.
Narrow font	Identifies storage locations for services on the webMethods Integration Server using the convention *folder.subfolder:service*.
`Typewriter font`	Identifies characters and values that you must type exactly or messages that the system displays on the console.
UPPERCASE	Identifies keyboard keys. Keys that you must press simultaneously are joined with the "+" symbol.
\	Directory paths use the "\" directory delimiter unless the subject is UNIX-specific.
[ ]	Optional keywords or values are enclosed in [ ]. Do not type the [ ] symbols in your own code.

Additional Information

The webMethods Advantage Web site at http://advantage.webmethods.com provides you with important sources of information about the webMethods Integration Server:

- **Troubleshooting Information.** webMethods provides troubleshooting information for many webMethods components in the webMethods Knowledge Base.
- **Documentation Feedback.** To provide documentation feedback to webMethods, go to the Documentation Feedback Form on the webMethods Bookshelf.
- **Additional Documentation.** All webMethods documentation is available on the webMethods Bookshelf.

webMethods.

CHAPTER 1

Adapter Services

Overview

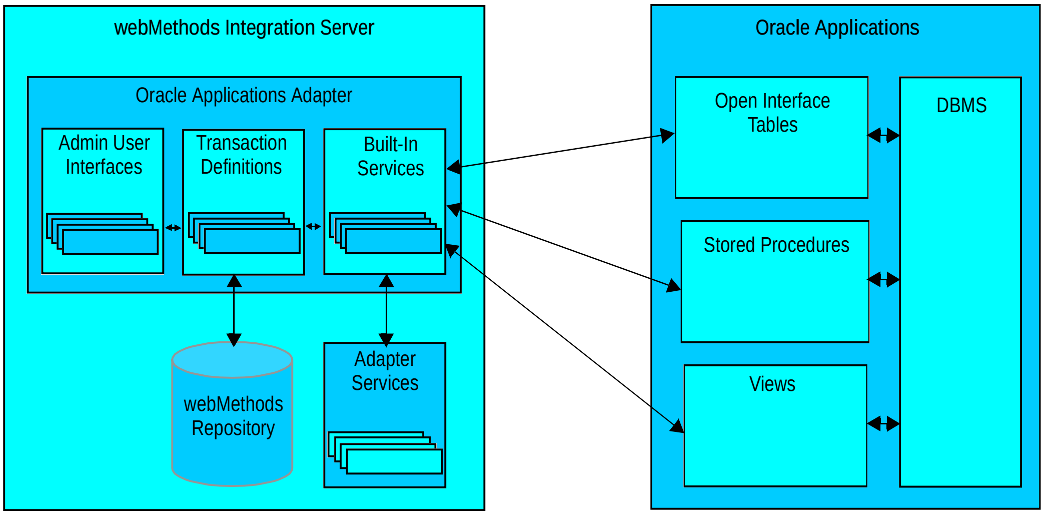
Adapter services enable you to integrate with Oracle Applications systems by executing transactions against Oracle Applications systems. Adapter services use transaction definitions to determine the processing to perform on an Oracle Applications system.

Sample transaction definitions are available that you can use to configure your own adapter services. You can use the samples as they are provided, or you can modify them or create new transaction definitions customized for your needs. This flexibility facilitates integration with highly customized Oracle Applications.

Transaction definitions are interfaces to Oracle Applications functions. The sample transaction definitions wrap the functionality of open interface tables, stored procedures, views, and raw SQL statements. For example, there is a sample transaction definition that wraps the Sales Order open interface table in the Oracle Applications Financials package. The transaction definitions, along with the Oracle Applications Adapter connection definitions, are stored in the webMethods repository.

Important! For Oracle Applications integrations that are not highly customized, there are sets of predefined transaction services that are available as add-ons to the webMethods Oracle Applications Adapter. Predefined transaction services are complete, ready-to-use flow services that simplify the process of integrating with Oracle Applications systems. We recommend that if there are predefined transaction services that perform the operation you want to accomplish using a adapter services you created, you should use the predefined transaction services rather than creating your own adapter services.

The following diagram shows in more detail how the Oracle Applications Adapter works with adapter services.



- Oracle Applications Adapter. The Oracle Applications Adapter is delivered in a package named WmOAAdapter. The WmOAAdapter package includes:
  - Administrative user interfaces to configure and manage adapter connections and transaction definitions. Using these interfaces you can:
    - Perform administrative tasks.
    - Configure the login parameters that the Oracle Applications Adapter uses to establish a client connection with one or more Oracle Applications.
  - Developer user interface to configure an adapter service for each transaction that you want to expose (make available) through the webMethods Integration Server.
  - Built-in services that provide the basic means to interact with Oracle Applications.

    The built-in services perform activities such as inserting and updating data, and managing database connections. Adapter services wrap built-in functions and make them accessible to the webMethods Integration Server.
- Open Interface Tables. Rather than directly updating the Oracle Applications DBMS, the adapter services first place the data into open interface tables. Oracle then runs an import process to copy the data from the tables into the live Oracle Applications database. Data is checked during table insertion and again during the import process, thus ensuring data integrity.
- Stored Procedures. webMethods provides sample transaction definitions that map to the stored procedures. You configure adapter services from these transaction definitions to programmatically retrieve data from the DBMS.
- Views. webMethods provides views that you install on the Oracle Applications instance. Oracle Applications uses the views to perform webMethods transactions.
- DBMS. The Oracle database used to power the Oracle Applications product.

Transaction Definitions

Transaction definitions map to functions you can perform in Oracle Applications systems. Use the Oracle Applications Adapter to configure and run adapter services, which execute transactions to retrieve data from, and insert and update data in, Oracle Applications systems via the Oracle Applications open interface tables.

There are sample transaction definitions for Oracle Applications versions 10.7SC, 11.0, and 11.5. See the following chapters for descriptions of the sample transaction definitions.

Chapter	Page
Chapter 2, "Transaction Definitions (IS-to-Oracle Applications 11.5)"	31
Chapter 3, "Transaction Definitions (IS-to-Oracle Applications 11.0)"	49
Chapter 4, "Transaction Definitions (Oracle Applications 11.0-to-IS)"	69
Chapter 5, "Transaction Definitions (IS-to-Oracle Applications 10.7SC)"	97
Chapter 6, "Transaction Definitions (Oracle Applications 10.7SC-to-IS)"	115

A transaction definition can map to open interface tables, stored procedures, views, and raw SQL statements. The Oracle Applications Adapter enables you to modify the sample transaction definitions and to define your own transaction definitions. This flexibility facilitates integration with highly customized Oracle Applications. For more information about configuring or editing transaction definitions, see the *webMethods Oracle Applications Adapter User's Guide*.

Note: The sample transaction definitions described in this guide are the same transaction definitions that were provided with the Oracle Applications Adapter version 2.0.

Importing and Exporting Transaction Definitions

The sample transaction definitions are provided as .txp files, and are located in the following directory, where *nnn* is the version of Oracle Applications:

IntegrationServer_Directory\packages\WmOAAdapter\OriginalTxns_*nnn*

The transaction definitions are delivered in the following files:

Transaction Set	File Name
Oracle Applications 11.5	wm_115.txp
Oracle Applications 11.0	wm_110.txp
Oracle Applications 10.7SC	wm_107SC.txp

Before you can use the sample transaction definitions you must import them into your webMethods system.

You can import transaction definition files from the Server Administrator or from the Developer. For instructions, see the chapter on managing transactions in the *webMethods Oracle Applications Adapter User's Guide*. After you import the transaction definitions, they are stored in the webMethods repository.

Note: If you are upgrading from an earlier version of the adapter to version 6.0, you must re-import the transaction definitions into the repository.

The adapter also enables you to export your transaction definitions to files. You can export your transactions to files to back up your transaction definitions or to transfer them to a different installation of the Oracle Applications Adapter. See the *webMethods Oracle Applications Adapter User's Guide* for instructions for importing and exporting transaction definitions.

Adapter Services

You use the Oracle Applications Adapter to configure adapter services from transaction definitions. The services you configure perform the actions that the transaction definitions define. In other words, adapter services:

- Interact with open interface tables to update the Oracle DBMS.
- Execute a stored procedure or view to retrieve or update information in the Oracle DBMS.
- Execute the SQL SELECT to retrieve information from the Oracle DBMS.
- Execute the SQL UPDATE to modify information in the Oracle DBMS.

You can invoke the services that you create from any type of webMethods client: browser client, Java client, C/C++ client, or Visual Basic client. For example, you can create a browser client that allows a user to fill in an HTML form that interacts with your Oracle Applications instance via the Oracle Applications Adapter.

You can create a webMethods client that invokes the adapter service, or you can integrate the adapter service into an existing flow service. For example, if you want to code a webMethods client that creates a customer order from a purchase order, code the webMethods client to invoke the service that executes the OEOrderImport transaction of Oracle Order Management.

Database Scripts

Adapter services require some components to be installed on the Oracle Applications system. These components include items such as stored procedures and views. You install these components using database scripts.

After you install the Oracle Applications Adapter and import the set of transaction definitions for the version of Oracle Applications you are using, you must run the appropriate database scripts to set up the Oracle Applications database for use with the adapter services.

Installing Service Components on the Oracle Applications Database

To set up the Oracle Applications database for use with adapter services

1 Copy files from the appropriate directory as follows:

- If you are using Oracle Applications 11.5, copy all files from the *IntegrationServer_Directory*\packages\WmOAAdapter\sampleTxns\setup\11.5 directory into the bin directory of your Oracle database (orant\bin).
- If you are using Oracle Applications 11.0, copy all files in the *IntegrationServer_Directory*\packages\WmOAAdapter\sampleTxns\setup\11.0 directory into the bin directory of your Oracle database (orant\bin).
- If you are using Oracle Applications 10.7SC, copy all files in the *IntegrationServer_Directory*\packages\WmOAAdapter\sampleTxns\setup\10.7SC directory into the bin directory of your Oracle database (orant\bin).

2 Log in to SQL*Plus. You can use any login.

3 From SQL*Plus, run the installation script:

```
@wmOASetup.sql
```

When prompted, supply the APPS username/password, a Custom username/password, custom tablespaces (for data and index), and a connection string.

Note: Make sure you enter the APPS user and its associated password. In addition, the procedures and views being created must be in the APPS schema.

When the installation is complete, you will see the message “Installation completed.”

4 Exit SQL*Plus.

Note: If you install the setup files in a schema other than APPS (for Oracle 11.0) or APPS_APPDEMO (for Oracle 10.7), you must update the following Oracle Applications-to-IS transaction definitions:

- InvoiceOutbound
- InvoiceOutbound_Ack
- PurchaseOrderNew_Ack
- PurchaseOrderChange_Ack
- AdvanceShipNotice_Ack (10.7 only)

These transaction definitions call stored procedures with the schema.package.procedure format. You must edit the schema on these calls to match your installation. For more information about editing transaction definitions, see the *webMethods Oracle Applications Adapter User's Guide*.

Uninstalling Service Components from the Oracle Applications Database

If you uninstall the adapter or no longer want to use a set of adapter services, you should uninstall the services' components from the Oracle Applications database.

To remove adapter service objects from the Oracle Applications database

1. Log in to SQL*Plus. You can use any login.
2. From SQL*Plus, run the removal script:

   ```
   @wmOARemove.sql
   ```

   When prompted, supply the APPS username/password, a Custom username/password, custom tablespaces (for data and index), and a connection string.

   When the uninstallation is complete, you will see the message “Removal completed.”
3. Exit SQL*Plus.

Built-In Utility Services

The Oracle Applications Adapter provides two built-in utility services. These services are provided to help you more conveniently manage multiple transaction definitions at a time, rather than having to manage them individually.

- The pub.adapter.wmoa:deleteAllTransactions service enables you to delete all transaction definitions from the Integration Server repository at once for a given version of Oracle Applications. Using this service lets you use the Developer rather than deleting transaction definitions individually using the adapter's Integration Server Administrator screens.
- The pub.adapter.wmoa:importAllTransactions service enables you to import all transaction definitions using the Developer rather than importing them individually using the adapter's Integration Server Administrator screens.

Note: The only built-in services supported for direct use are the utility services provided in the pub.adapter.wmoa folder in the WmOAAdapter package within Developer. Other services that appear in the WmOAAdapter package are used internally by the adapter.

For more information about the adapter's built-in services, see Chapter 7, "Oracle Applications Adapter Built-In Utility Services" on page 147. See the *webMethods Oracle Applications Adapter User's Guide* for instructions on how to import and delete transaction definitions.

Transaction Processing

The adapter communicates using two types of transactions: Oracle Applications-to-IS transactions and IS-to-Oracle Applications transactions. Within each type of transaction there are two types of operations. For IS-to-Oracle Applications transactions, the adapter can insert data into open interface tables and can execute error SQL to retrieve error information. For Oracle Applications-to-IS transactions, the adapter can query the Oracle Applications system and can run corresponding acknowledgement (ACK) services, which mark the data as processed.

The following sections describe the processing for these types of transactions.

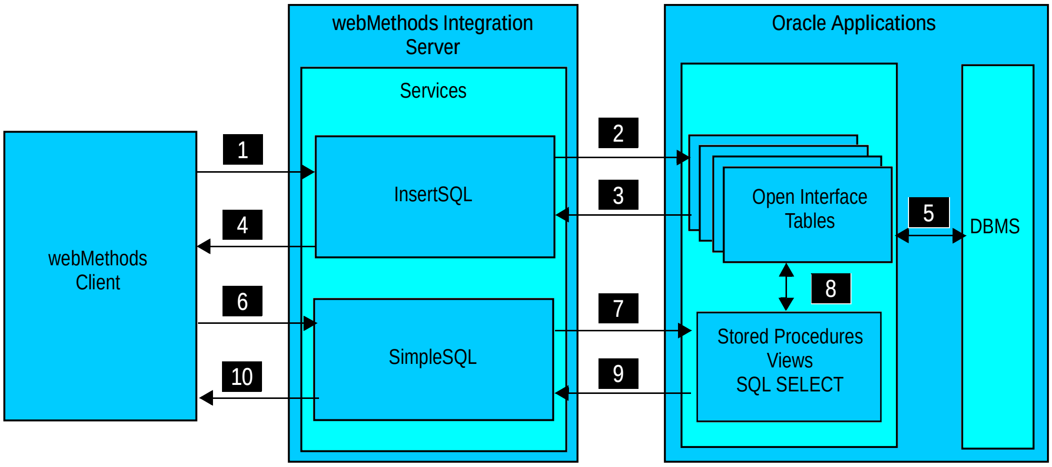
webMethods IS-to-Oracle Applications (Inbound)

webMethods clients can invoke adapter services that insert records into open interface tables. These records will either insert new information or update existing information in the Oracle system. webMethods clients can also invoke services to retrieve open interface import errors from the Oracle system.

To retrieve information from the Oracle database, a service does one of the following:

- Executes stored procedures on the Oracle system
- Executes an SQL SELECT statement

The following diagram illustrates the steps that occur when you run a service to update the Oracle DBMS and then run a service to check for errors that might have occurred during import processing.



Step	Description
1	The webMethods client invokes a service to update the Oracle DBMS. You created this service from a transaction definition that is based on a set of specific open interface tables. The service receives the information required by the open interface table via the service specification.
2	The service executes. It connects to the Oracle Applications Instance via JDBC and inserts data into the open interface tables that are associated with the service.
3	The service receives output to indicate that the table insert is complete.
4	Output is returned to the client.
5	An Oracle Applications import process runs in the Oracle concurrent manager. The import process moves the data from the open interface tables to the live Oracle DBMS. If errors occur in the import process, Oracle records them in the appropriate open interface table.

Step	Description
6	The webMethods client invokes a service to query Oracle Applications for errors that might have occurred during the import process. You created this service from a transaction definition that is based on a stored procedure, view, or SQL SELECT statement. The service receives the query parameters for the SQL statement via the service specification. The service can also be invoked by the webMethods scheduler to periodically pull out transactions for a trading partner.
7	The service executes. It connects to the Oracle Applications Instance via JDBC. Based on the transaction definition used to create the service, the service does one of the following: ■ Calls a stored procedure ■ Executes an SQL SELECT statement
8	The stored procedure or SQL SELECT statement retrieves information from the DBMS.
9	The retrieved information is returned to the service.
10	The service returns output back to the webMethods client.

How Data Is Passed During the Transaction

When a webMethods client invokes an adapter service, it passes input values to the service in a document. The service then passes the data from the document in an SQL statement. The statement is executed via a JDBC connection.

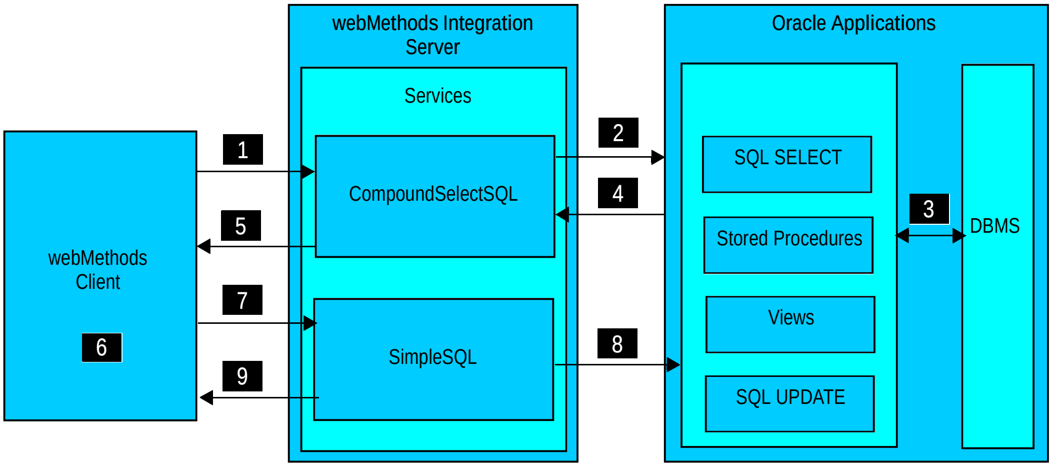
When a service successfully inserts data into the open interface table, the service puts a variable called *message* into the pipeline as output. If there is a problem inserting the data into the Oracle Applications open interface tables, a service exception is thrown.

Services that retrieve information from the Oracle system return a document that contains the requested data. If there is a problem selecting the information, a service exception is thrown.

Oracle Applications-to-webMethods IS (Outbound)

webMethods clients can invoke adapter services that retrieve information from the Oracle Applications system. Generally, the Oracle Applications Adapter initiates the interaction with the Oracle Applications instance. However, you can use the Oracle Applications Adapter retrieve the business document to update other systems (legacy systems, ERP systems, or trading partner systems) with information retrieved from the Oracle system. Clients can also invoke acknowledgment services to maintain an accurate status of the transactions within the Oracle Applications system.

The following diagram illustrates the steps required to use Oracle information to update a trading partner's system.



Step	Description
1	The webMethods client invokes a service that you created from a transaction definition that is based on a stored procedure, view, or SQL SELECT statement. The service passes query parameters to the stored procedure, view, or SQL SELECT statement.
2	The service executes. It connects to the Oracle Applications Instance via JDBC. Based on the transaction definition used to create the service, the service does one of the following: ■ Calls a stored procedure ■ Executes an SQL SELECT statement against tables and views.
3	The stored procedure or SQL SELECT statement retrieves information from the DBMS.
4	The retrieved information is returned to the adapter service.
5	The service returns the output to the webMethods client.
6	The webMethods client optionally performs internal processing or invokes services. For example, you might choose to update your SAP system at this stage.

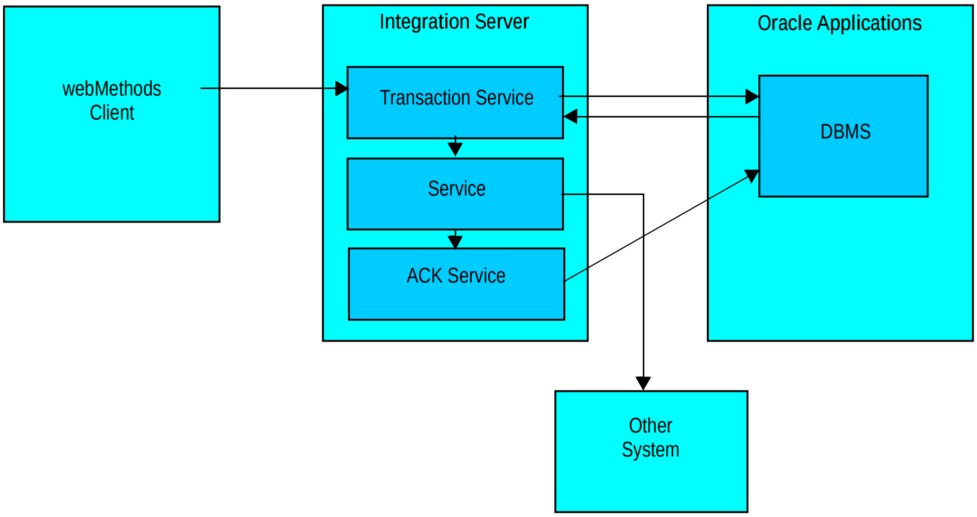
Step	Description
7	If there is an acknowledgement service defined for the transaction, the flow needs to execute it. The acknowledgement service connects to the Oracle Applications instance via JDBC. Based on the transaction definition used to create it, the service does one of the following: - Calls a stored procedure - Executes an SQL UPDATE statement **Note:** The acknowledgement service is necessary for Oracle Applications to maintain the accurate status of transactions. For example, suppose that a purchase order has state of "approved" in the DBMS. It remains "approved" until an acknowledgement from the client is received, which changes the state to "approved, printed." This ensures that the purchase order, which has already been received by the client, will not be included in the next data retrieval transaction.
8	The adapter service tells the Oracle Applications instance that the transaction has been processed.
9	The service returns output back to the webMethods client.

Considerations When Using Transactions

Considerations When Using Oracle Applications-to-IS Transactions

- Oracle Applications-to-IS transactions perform selects of information that is stored in the Oracle Applications DBMS. It can be selected for viewing, in which case the service can be invoked and no acknowledgement needs to be done.
- More often the transaction will be invoked from a flow service that processes the business transaction (such as sending it to another business).
- After the transaction is successfully processed, your flow needs to call the acknowledgement service. The acknowledgement service marks the corresponding data (for example a PO, Invoice) as processed, ensuring that the same data will not be returned in subsequent calls to the service.
- Within a flow, you can map values to a service's queryParams input record, and to the dataRecord input record for the service's associated acknowledgement service.

The following diagram shows at a high level the Oracle Applications-to-IS transaction processing when another service is invoked to send a business transaction to another system:



Considerations When Using IS-to-Oracle Applications Transactions

- IS-to-Oracle Applications transactions defined with the Oracle Applications Adapter insert data into the Oracle Applications open interface tables.
- Errors that occur while data is inserted into the open interface tables insertion are reported when the service executes.
- After an IS-to-Oracle Applications transaction has executed successfully, you should schedule a call to the Error Log Service (created at the same time as the Oracle Applications transaction service) to make sure no errors occurred during the Oracle Applications import process.

  Alternatively, you can publish the Error Log Service or a web page that calls it so the partner that sent the transaction can check if there is an error.
- Many Oracle Applications transactions contain a CREATED_BY column. If you want to track transactions inserted by the Oracle Applications Adapter, create an Oracle Applications user and set the CREATED_BY input column with that user name.
- Within a flow, you can map values to a service's dataRecord input record, and to the queryParams input record for the service's associated error service.

Considerations for Multiple Organizations

You can configure a single installation of Oracle Applications to support multiple organizations within an enterprise.

- Oracle Applications-to-IS transactions perform selects of information that is stored in the Oracle Applications DBMS. These select statements and stored procedures work on views defined on the Oracle Applications DBMS. Transaction definitions are available for use with the Oracle Applications Adapter that return data from all organizations within the Oracle Applications instance. You can customize these views or the queries in the transaction definition by adding a WHERE clause that subsets the data returned to a particular organization ID or set of organization IDs.
- IS-to-Oracle Applications transaction definitions insert data into the Oracle Applications open interface tables. The webMethods Integration Server can receive transactions from any number of sources, via any number of protocols.
- When you implement integration between two trading partners, your service must be able to figure out who the transaction is for; in other words, it must be able to map organization IDs. When your service encounters an organization ID, it needs to look the ID up in a database, in an LDAP store, or it must be hard-coded in the service.

Considerations for webMethods Clients

Use the information in this section when creating a webMethods client that invokes a service that executes an Oracle Applications transaction. It describes:

- How to invoke a service that executes an Oracle Applications transaction.
- Error conditions that you should be aware of when creating a webMethods client.

Generating Client Code

You can use the Developer to generate Java client code for services that execute an Oracle Applications transaction in the same manner that you generate Java client code for any other service. However, you must make modifications to the Java if the service expects any input variables that have a complex type (that is, sequences or structures). The Developer does not generate code to prompt for complex input variables. You will need to update your client Java file to prompt for the complex inputs or build the inputs programmatically.

Refer to the *webMethods Developer User's Guide* for instructions on how to create client code for a service

Determining the Input and Output for the Service

Because the input to and output from the service has a specification, you can use the Developer to examine the specification of the service to learn more about the required input and output.

Invoking a Service from a Client

You invoke a service created from an Oracle Applications transaction in the same manner that you invoke any other service. Refer to the *webMethods Developer User's Guide* for instructions on how to invoke a service.

Handling Error Conditions

This section identifies some reasons a service might fail. Code your webMethods client to handle these errors.

- If an error occurs during an Oracle Applications-to-IS transaction, the Oracle Applications Adapter throws a service exception.
- If an error occurs on an IS-to-Oracle Applications transaction during insertion into the Oracle open interface tables, the Oracle Applications Adapter throws a service exception.
- The Oracle Applications Adapter provides a service to get error details if an error occurs on an IS-to-Oracle Applications transaction during processing by the Oracle Application import process.
- Errors are logged in the webMethods Integration Server's log files. You can view the log files using the webMethods Administrator. Refer to the *webMethods Integration Server Administrator's Guide* for instructions.

JDBC errors—If the transaction fails for any reason, it returns the error information to the Oracle Applications Adapter. The service returns the error information to the client in a document. Most errors that can be returned by a JDBC application can be returned by the Oracle Applications Adapter.

Oracle Applications errors—If an IS-to-Oracle Applications transaction is successfully inserted into the Oracle Applications open interface tables, the transaction service will return success. A service is provided to return transaction status that includes customizable details on Oracle Applications errors for each specific transaction.

Database Connection Management

Logging Into and Out of an Oracle Applications Instance

The webMethods Integration Server logs into the Oracle Applications instance automatically upon starting up a connection that you configured using the Oracle Applications Adapter. The Integration Server logs out of the Oracle Applications instance automatically when the session terminates.

For information about configuring Oracle Applications Adapter connections, see the *webMethods Oracle Applications Adapter User's Guide*.

CHAPTER 2

Transaction Definitions (IS-to-Oracle Applications 11.5)

Overview

This chapter provides an in-depth description of the IS-to-Oracle Applications 11.5 sample transaction definitions. IS-to-Oracle transactions update the Oracle database with information from an end-user or XML file.

This chapter describes the transactions as they exist in the samples. It does not cover any modifications that you can make to the sample transactions.

Transaction Overview

The sample transactions enable you to use the Oracle Applications Adapter to create adapter services to insert data into Oracle Applications open interface tables. A fully defined IS-to-Oracle transaction has two parts: the open interface table(s) and an error SQL query. The transaction may group one or more open interface tables together. Every column from each open interface table for the defined transaction is exposed as an input to the adapter service. After the service inserts records into the corresponding open interface tables, Oracle Applications must run the respective import process.

After a transaction completes, you can test it for errors from the webMethods Integration Server by using the error SQL query. The SQL query has inputs into which you can map query parameter values. The service then runs the query and the results are available as service outputs.

For more information about the IS-to-Oracle transactions, see your Oracle Applications documentation.

This chapter lists the transactions in alphabetical order. The table below shows the transactions organized by Oracle Applications module.

Module	webMethods Transaction	Page
Payables	"APInvoices"	33
Receivables	"ARAutoInvoice"	35
	"ARCustomers"	36
Inventory	"INVCustomerItemCrossRef"	38
	"INVCustomerItems"	39
	"INVItems"	40

Module	webMethods Transaction	Page
Order Management	"OEOrderImport"	41
Purchasing	"PODocuments"	43
	"POReceiving"	44
	"PORequisitions"	45
	"POResheduleRequisitions"	46

APInvoices

Use this transaction to create invoices for existing purchase orders. The Payables Open Interface Import program creates invoices to import into Payables, using invoice data stored in the Payables open interface tables.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Payables

Open Interface

Payables Open Interface

Open Interface Tables

The open interface tables exposed by this transaction are:

- AP_INVOICES_INTERFACE
- AP_INVOICE_LINES_INTERFACE

Error SQL

Inputs

Field	Description
SOURCE	A value to match against the SOURCE column in a corresponding AP_INVOICES_INTERFACE row. This parameter will accept wild cards. % returns all errors.

Outputs

Field	Description
PO_NUMBER	Purchase order number associated with the Invoice.
VENDOR_ID	Vendor ID.
VENDOR_NUM	Vendor number.
VENDOR_NAME	Vendor name.
VENDOR_SITE_ID	Vendor site ID.
VENDOR_SITE_CODE	Vendor site code.
INVOICE_NUM	Invoice number.
INVOICE_ID	Invoice ID.
LINE_NUMBER	Invoice line number.
DESCRIPTION	Error code description.
DISPLAYED_FIELD	Error code display value.

ARAutoInvoice

Use this transaction to create invoices, debit memos, credit memos, and on-account credits in Oracle Receivables.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Receivables

Open Interface

AutoInvoice

Open Interface Tables

The open interface tables exposed by this transaction are:

- RA_INTERFACE_LINES_ALL
- RA_INVOICE_SALESCREDITS_ALL
- RA_INTERFACE_DISTRIBUTION_ALL

Error SQL

Inputs

Field	Description
WM_USER	Name of the user for the corresponding user ID in the CREATED_BY column of the RA_INTERFACE_LINES_ALL table. This parameter will accept wild cards. % returns all errors.

Outputs

Field	Description
INTERFACE_LINE_ID	Contains the INTERFACE_LINE_ID from the RA_INTERFACE_LINES_ALL table.
INTERFACE_SALESCREDIT_ID	Contains the INTERFACE_SALESCREDITS_ID from the RA_INVOICE_SALESCREDIT_ALL table. This field will be empty if the error is from a different table.
INTERFACE_DISTRIBUTION_ID	Contains the INTERFACE_DISTRIBUTION_ID from the RA_INVOICE_DISTRIBUTIONS_ALL table. This field will be empty if the error is from a different table.
MESSAGE_TEXT	Error message.
INVALID_VALUE	The value that caused this error.
LINK_TO_LINE_ID	Contains the INTERFACE_LINE_ID from the associated row in the RA_INTERFACE_LINES_ALL table. A value will be in this field if the error occurred in the SALESCREDIT or DISTRIBUTIONS tables.
ORG_ID	Operating Unit Identifier.

ARCustomers

Use this transaction to create, import, and validate customers, customer profiles, addresses, contacts, and phone numbers from other systems.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Receivables

Open Interface

Customer Transaction

Open Interface Tables

The open interface tables exposed by this transaction are:

- RA_CUSTOMERS_INTERFACE_ALL
- RA_CUSTOMER_BANKS_INT_ALL
- RA_CUST_PAY_METHOD_INT_ALL
- RA_CONTACT_PHONES_INT_ALL
- RA_CUSTOMER_PROFILES_INT_ALL

Error SQL

Inputs

Field	Description
WM_USER	Name of the user for the corresponding user ID in the CREATED_BY column of the interface table. This parameter will accept wild cards. % returns all errors.
ORIG_SYSTEM_CUSTOMER_REF	Original system identifier loaded in the ORIG_SYSTEM_CUSTOMER_REF column of the interface table. This parameter will accept wild cards. % returns all errors.

Outputs

Field	Description
ERROR_EXISTS	Returns a 1 if an error is found in any of the corresponding interface tables; otherwise, it returns 0.

INVCustomerItemCrossRef

Use this transaction to import cross-references between customer Items and existing Oracle Inventory Items into your master Organization.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Inventory

Open Interface

Customer Item Cross Reference

Open Interface Tables

The open interface table exposed by this transaction is:

- MTL_CI_XREFS_INTERFACE

Error SQL

Inputs

Field	Description
WM_USER	Name of the user for the corresponding user ID in the CREATED_BY column of the MTL_CI_XREFS_INTERFACE table. This parameter will accept wild cards. % returns all errors.

Outputs

Field	Description
CUSTOMER_ ITEM_NUMBER	Customer item number in legacy system.
INVENTORY_ITEM_ID	Existing inventory Items ID.
TRANSACTION_TYPE	Transaction type (CREATE or UPDATE).
PROCESS_MODE	Process flag to indicate whether processing of row is succeeded or failed.

Field	Description
CUSTOMER_NAME	Customer name.
ERROR_EXPLANATION	Detailed error message.

INVCustomerItems

Use this transaction to import customer items from your legacy system into Oracle Inventory. For each customer item, you must define related information such as the Customer and Item Definition Level.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Inventory

Open Interface

Customer Item

Open Interface Tables

The open interface table exposed by this transaction is:

- MTL_CI_INTERFACE

Error SQL

Inputs

Field	Description
WM_USER	Name of the user for the corresponding user ID in the CREATED_BY column of the MTL_CI_INTERFACE table. This parameter will accept wild cards. % returns all errors.

Outputs

Field	Description
CUSTOMER_ ITEM_NUMBER	Customer item number in legacy system.
CUSTOMER_ITEM_DESC	Description of item.
TRANSACTION_TYPE	Transaction type (CREATE or UPDATE).
PROCESS_MODE	Process flag to indicate whether processing of row has succeeded or failed.
CUSTOMER_NAME	Customer name.
ERROR_EXPLANATION	Detailed error message.

INVItems

Use this transaction to convert inventory items from another inventory system, migrate assembly and component items from a legacy manufacturing system, convert purchased items from a custom purchasing system, and insert new items from a Product Data Management package.

When you import items using this transaction, it creates new Items in the item master organization or assigns existing Items to additional organizations. You can specify values for all item attributes, or you can use default or null values. This transaction also lets you import revision details, including past and future revisions and effective dates.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Inventory

Open Interface

Items Transaction

Open Interface Tables

The open interface tables exposed by this transaction are:

- MTL_SYSTEM_ITEMS_INTERFACE
- MTL_ITEM_REVISIONS_INTERFACE

Error SQL

Inputs

Field	Description
WM_USER	Name of the user for the corresponding user ID in the CREATED_BY column of the MTL_SYSTEM_ITEMS_INTERFACE table. This parameter will accept wild cards. % returns all errors.

Outputs

Field	Description
ITEM_NUMBER	Item number in legacy system.
DESCRIPTION	Description of item.
TRANSACTION_TYPE	Transaction type (CREATE or UPDATE).
PROCESS_FLAG	Process flag to indicate the processing state of the row.
ERROR_MESSAGE	Detailed error message.

OEOrderImport

Use this transaction to import order entry data from a variety of sources. You can import data into orders with lines, schedule details, price adjustments, and sales orders. You can import complete, incomplete, and booked orders. You can also import quota, or non-quota sales credit information, changes to existing orders, line schedule detail information with each order and order line, and internal requisition orders from Oracle Purchasing.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Order Management

Open Interface

Order Import

Open Interface Tables

The open interface tables exposed by this transaction are:

- OE_HEADERS_IFACE_ALL
- OE_LINES_IFACE_ALL
- OE_PRICE_ADJS_IFACE_ALL
- OE_CREDITS_IFACE_ALL
- OE_LOTSERIALS_IFACE_ALL

Error SQL

Inputs

Field	Description
WM_USER	Name of the user for the corresponding user ID in the CREATED_BY column of the OE_HEADERS_IFACE_ALL table. This parameter will accept wild cards. % returns all errors.
ORIG_SYS_DOCUMENT_REF	Original system identifier loaded in the ORIG_SYS_DOCUMENT_REF column of the OE_HEADERS_IFACE_ALL table. This parameter will accept wild cards. % returns all errors.

Outputs

Field	Description
ORIG_SYS_DOCUMENT_REF	Original system reference from the external system.
MESSAGE_TEXT	Error message.
CUSTOMER_PO_NUMBER	Customer purchase order related to the sales order.
CUSTOMER_NAME	Customer name associated with the order.

PODocuments

Use this transaction to import price and sales catalog information and responses into requests for quotations from suppliers.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Purchasing

Open Interface

Purchasing Documents

Open Interface Tables

The open interface tables exposed by this transaction are:

- PO_HEADERS_INTERFACE
- PO_LINES_INTERFACE

Error SQL

Inputs

Field	Description
WM_USER	Name of the user for the corresponding user ID in the CREATED_BY column of the PO_HEADERS_INTERFACE table. This parameter will accept wild cards. % returns all errors.

Outputs

Field	Description
INTERFACE_TYPE	Error message source.
INTERFACE_TRANSACTION_ID	Error transaction unique identifier.
COLUMN_NAME	Error column name.

Field	Description
ERROR_MESSAGE	Error message text.
PROCESSING_DATE	
INTERFACE_HEADER_ID	
INTERFACE_LINE_ID	

POReceiving

Use this transaction to import receipt information from other Oracle Applications or non-Oracle systems. You use this interface to integrate your Oracle purchasing application with new or existing applications.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Purchasing

Open Interface

Receiving

Open Interface Tables

The open interface tables exposed by this transaction are:

- RCV_HEADERS_INTERFACE
- RCV_TRANSACTIONS_INTERFACE

Error SQL

Inputs

Field	Description
WM_USER	Name of the user for the corresponding user ID in the CREATED_BY column of the RCV_HEADERS_INTERFACE table. This parameter will accept wild cards. % returns all errors.

Outputs

Field	Description
INTERFACE_TRANSACTION_ID	Error transaction unique ID.
COLUMN_NAME	Error column name.
ERROR_MESSAGE	Error message text.
PO_HEADER_ID	Purchase order header ID.
PO_LINE_ID	Purchase order line ID.
VENDOR_NAME	Vendor name.

PORequisitions

Use this transaction to import requisitions from other Oracle Applications or non-Oracle systems. This transaction enables you to integrate your Oracle Applications quickly with new or existing applications such as material replenishment, planning, inventory management, and production control systems.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Purchasing

Open Interface

Requisitions

Open Interface Tables

The open interface tables exposed by this transaction are:

- PO_REQUISITIONS_INTERFACE_ALL
- PO_REQ_DIST_INTERFACE_ALL

Error SQL

Inputs

Field	Description
WM_USER	Name of the user for the corresponding user ID in the CREATED_BY column of the PO_REQUISITIONS_INTERFACE_ALL table. This parameter will accept wild cards. % returns all errors.

Outputs

Field	Description
INTERFACE_TYPE	Interface type that generated the error.
INTERFACE_TRANSACTION_ID	Error transaction unique ID.
COLUMN_NAME	Column in which the error occurred.
ERROR_MESSAGE	Detailed explanation of the error.
SOURCE_TYPE_CODE	The source for this requisition.
QUANTITY	Number of Items in the requisition.

POReschedule Requisitions

Use this transaction to reschedule requisitions according to changes in your planned orders.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Purchasing

Open Interface

Requisitions

Open Interface Tables

The open interface table exposed by this transaction is:

- PO_RESCHEDULE_INTERFACE

Error SQL

None.

webMethods.

CHAPTER 3

Transaction Definitions (IS-to-Oracle Applications 11.0)

Overview

This chapter provides an in-depth explanation of the IS-to-Oracle Applications 11.0 sample transaction definitions. IS-to-Oracle transactions update the Oracle database with information from an end-user or XML file.

This chapter describes the transactions as they exist in the samples. It does not cover any modifications that you can make to the sample transactions.

For information about Oracle Applications 11.0-to-IS transactions, see Chapter 4, "Transaction Definitions (Oracle Applications 11.0-to-IS)".

Transaction Overview

The sample transactions enable you to use the Oracle Applications Adapter to create adapter services to insert data into Oracle Applications open interface tables.A fully defined IS-to-Oracle transaction has two parts: the open interface table(s) and an error SQL query. The transaction may group one or more open interface tables together. Every column from each open interface table for the defined transaction will be exposed as input to the adapter service. After the service inserts records into corresponding open interface tables, Oracle Applications must run the respective import process. When completed, you can test the transaction for errors from the webMethods Integration Server by using the error SQL query. This has inputs into which the user maps query parameter values. The service then runs the query and the results are available as service outputs.

For more information on the IS-to-Oracle transactions, see your Oracle Applications documentation.

This chapter shows the transactions in alphabetical order. The table below shows the transactions organized by Oracle Applications module.

Module	webMethods Transaction	Page
Payables	"APInvoiceOpen"	51
Receivables	"ARCustomers"	52
Inventory	"INVCustomerItems"	54
	"INVCustomerItemsCrossReference"	56
	"INVOpenTransactions"	58
	"INVItems"	59

Module	webMethods Transaction	Page
Order Entry	"OEOrderImport"	61
Purchasing	"PODocuments"	63
	"POReceiving"	65
	"PORequisitions"	66
	"POResceduleRequisitions"	68

APInvoiceOpen

Use this transaction to create invoices for existing purchase orders. The Payables Open Interface Import program creates invoices to import into Payables, using invoice data stored in the Payables open interface tables.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Payables

Open Interface

AP Invoice Open Import Transaction

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- AP_INVOICES_INTERFACE
- AP_INVOICE_LINES_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- AP_INTERFACE_REJECTIONS
- AP_INVOICES_INTERFACE
- AP_LOOKUP_CODES
- AP_INVOICE_LINES_INTERFACE

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
PO_NUMBER	Purchase order number.
VENDOR_ID	Vendor ID.
VENDOR_NUM	Vendor number.
VENDOR_NAME	Vendor name.
VENDOR_SITE_ID	Vendor site ID.
VENDOR_SITE_CODE	Vendor site code.
INVOICE_NUM	Invoice number.
INVOICE_ID	Invoice ID.
LINE_NUMBER	Purchase order line number.
DESCRIPTION	Description.

Usage Comments

To import Items, run the Open Interface Invoices function, under Invoices, which is under the Payables module.

ARCustomers

Use this transaction to create and import customers, customer profiles, addresses, contacts, and phone numbers from other systems.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Receivables

Open Interface

Customer Transaction

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- RA_CUSTOMERS_INTERFACE_ALL
- RA_CUSTOMERS_BANKS_INT_ALL
- RA_CUST_PAY_METHOD_INT_ALL
- RA_CONTACT_PHONES_INT_ALL
- RA_CUSTOMER_PROFILES_INT_ALL

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- RA_CUSTOMERS_INTERFACE_ALL
- FND_USER

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.
CUSTOMER_NAME	Customer name to be imported.

Outputs

Field	Description
ORIG_SYSTEM_CUSTOMER_REF	Original system customer reference (that is, primary key for customer in other application).
CUSTOMER_NAME	Name of the customer to be imported.
INTERFACE_STATUS	Error codes. For error code definitions, see http://www.oracle.com/.

Field	Description
PROCESS_MODE	Process flag to indicate whether processing of row succeeded or failed.
CUSTOMER_NAME	Customer name.
ERROR_EXPLANATION	Detailed error message.

Usage Comments

- If Customer Interface fails because of the error SQL-02112 “Too Many Rows”, you must install Oracle Applications Patch 822300 (for Windows NT).
- To import invoice information, run the Customer Interface request, which is under the AR module.
- You can run Customer Interface Transfer report to see exact errors.
- `INACTIVE_FLAG` is a required field. Set it to “1” for Yes and “2” for No.

INVCustomerItems

Use this transaction to import customer items from any legacy system into Oracle Inventory. For each customer item, you must define related information such as the Customer and Item Definition Level. If you set Item Definition Level 3 while Customer Category is required for Item Definition Level 2, Customer Address is required.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Inventory

Open Interface

Customer Items Transaction

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- MTL_CI_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- MTL_CI_INTERFACE
- FND_USER_VIEW

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
CUSTOMER_ ITEM_NUMBER	Customer item number in old system.
INVENTORY_ITEM_DESC	Description of item.
TRANSACTION_TYPE	Transaction type (CREATE or UPDATE).
PROCESS_MODE	Process flag to indicate whether processing of row is succeeded or failed.
CUSTOMER_NAME	Customer name.
ERROR_EXPLANATION	Detailed error message.

Usage Comments

- To import Customer Items, run the **Import Customer Items** function, which is under the Inventory module.
- `PROCESS FLAG` and `PROCESS_MODE` columns report the status of the record after the import and validation process is complete. These columns are required on insert and should be populated with the value “1”.

- INACTIVE_FLAG is a required field. Set it to “1” for Yes and “2” for No.
- DEP_PLAN_REQUIRED_FLAG and DEP_PLAN_PRIOR_BLD_FLAG fields have required values of “1” for Yes and “2” for No.
- Oracle Applications does not allow update through this open interface. If you use a TRANSACTION_TYPE of UPDATE, the import will treat it as a CREATE and you receive a *Customer Item already exists* error.
- Oracle Applications does not allow deletion through this open interface. You may set a Customer Item to “Inactive” using the Oracle Applications GUI.

INVCustomerItemsCrossReference

Use this transaction to import cross-references between customer Items and existing Oracle Inventory Items into your master Organization. If you set Item Definition Level 3 while Customer Category is required for Item Definition Level 2, Customer Address is required.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Inventory

Open Interface

Customer Items Cross Reference

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- MTL_CI_XREFS_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- MTL_CI_XREFS_INTERFACE
- FND_USER_VIEW

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
CUSTOMER_ ITEM_NUMBER	Customer item number in old system.
INVENTORY_ITEM_ID	Existing inventory Items ID.
TRANSACTION_TYPE	Transaction type (CREATE or UPDATE).
PROCESS_MODE	Process flag to indicate whether processing of row is succeeded or failed.
CUSTOMER_NAME	Customer name.
ERROR_EXPLANATION	Detailed error message.

Usage Comments

- To create Customer Items Cross-References use a `TRANSACTION_TYPE` of `CREATE`.
- We have experienced a bug in Oracle Applications. If you set the **Delete Processed rows** option to **Yes**, the records should be removed from the open interface tables after the import. However, the records are not removed and are not updated to show they have been processed.
- `*PROCESS FLAG` and `PROCESS_MODE` columns report the status of the record after the import and validation process is complete. These columns are required on insert and should be populated with the value “1”.
- `INACTIVE_FLAG` is a required field. Set it to “1” for Yes and “2” for No.
- Oracle Applications does not allow update through this open interface. If you use a `TRANSACTION_TYPE` of UPDATE, the import will treat it as a CREATE and you receive a **Customer Item Cross Reference** already exists error.
- Oracle Applications does not allow deletion through this open interface. You may set a Customer Item Cross Reference to Inactive using the Oracle Applications GUI.
- To import Items, run the **Import Customer Items Cross Reference** function, which is under the Inventory module.

INVOpenTransactions

Use this transaction to load transactions from external applications and feeder systems, which include sales, order shipment transactions from an order entry system other than Oracle Order Entry, simple material issues, receipts, or transfers.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Inventory

Open Interface

Inventory Open Transaction

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- MTL_TRANSACTIONS_INTERFACE
- MTL_TRANSACTION_LOTS_INTERFACE
- MTL_SERIAL_NUMBERS_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- MTL_TRANSACTIONS_INTERFACE
- FND_USER_VIEW

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
INVENTORY_ITEM_ID	Inventory item ID.
TRANSACTION_QUANTITY	Number of Items in the transaction.
ERROR_EXPLANATION	Detailed error explanation.
TRANSACTION_SOURCE_NAME	Name of the source.

Usage Comments

To import transactions, run the **Import Open Transactions** function, which is under the Inventory module.

INVItems

Use this transaction to convert inventory items from another inventory system, migrate assembly and component items from a legacy manufacturing system, convert purchased items from a custom purchasing system, and insert new items from a Product Data Management package. When you import items using the INVItems transaction, it creates new Items in the item master organization or assigns existing Items to additional organizations. You can specify values for all item attributes, or you can use default or null values. Also, the INVItems transaction lets you import revision details, including past and future revisions and effective dates.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Inventory

Open Interface

Items Transaction

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- MTL_SYSTEM_ITEM_INTERFACE
- MTL_ITEM_REVISIONS_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- MTL_SYSTEM_ITEM_INTERFACE
- MTL_INTERFACE_ERRORS
- FND_USER_VIEW

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
ITEM_NUMBER	Item number in old system.
DESCRIPTION	Description of item.
TRANSACTION_TYPE	Transaction type (CREATE or UPDATE).
PROCESS_FLAG	Process flag to indicate whether processing of row has succeeded or failed. Possible process flags are: 1 Pending 2 Assign complete 3 Assign/validation failed 4 Validation succeeded; import failed 5 Import in process 6 Import succeeded
ERROR_MESSAGE	Detailed error message.

Usage Comments

- To create Items, choose TRANSACTION TYPE as CREATE. When importing these Items, choose Create Items.
- To update Items, choose TRANSACTION_TYPE as UPDATE. When importing these Items choose Update existing Items.
- If you have list of transactions containing both UPDATES and CREATES, give all CREATES one SET_PROCESS_ID and all UPDATES a different SET_PROCESS_ID. At the time of importing you specify Items to import by their SET_PROCESS_ID.
- Be careful when running the import process. We have experienced a problem running Oracle Applications 11.0 on Windows NT. If you specify a SET_PROCESS_ID used for your CREATE records and then during the import you specify the SET_PROCESS_ID for UPDATE records, a Dr. Watson Error will be triggered on Oracle Applications. The process flag on the records will show pending and the SET_PROCESS_ID will be set to –999.
- Oracle Applications does not allow the deletion of Items using the open interface tables. CREATE and UPDATE transaction types are currently supported.
- To import Items, run the Import Items function, which is under the Inventory module.

OEOrderImport

Use this transaction to import order entry data from a variety of sources. You can import data into orders with lines, schedule details, price adjustments, and sales orders. You can import complete, incomplete, and booked orders. You can also import quota, or non-quota sales credit information, changes to existing orders, line schedule detail information with each order and order line, and internal requisition orders from Oracle Purchasing.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Order Entry/Shipping

Open Interface

Order Import

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- SO_HEADERS_INTERFACE_ALL
- SO_LINES_INTERFACE_ALL
- SO_LINES_DETAIL_INTERFACE
- SO_SALES_CREDITS_INTERFACE
- SO_PRICE_ADJUSTMENTS_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- SO_HEADERS_INTERFACE_ALL
- FND_USER

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.
ORIG_SYSTEM_REF	Original system reference.
PO_NUMBER	Purchase order number.
CUSTOMER_NAME	Customer name.
CUSTOMER_NUMBER	Customer number in legacy system.

Outputs

Field	Description
ORIGINAL_SYSTEM_REFERENCE	Original system reference.
PURCHASE_ORDER_NUM	Purchase order number.
CUSTOMER_NUMBER	Customer number in legacy system.
CUSTOMER_NAME	Customer name.
INTERFACE_STATUS	Current status of the transaction (pending, rejected, or success).

Usage Comments

- For each record in the Line Interface, a corresponding record should exist in the Headers Table. Otherwise, the Lines record will be left dangling and will not be picked up by Import Process.
- You can enter Header Information without specifying Lines Detail.
- To import orders, run the Import Sales Order function, which is under the Order Entry module.

PODocuments

Use this transaction to import price and sales catalog information and responses into requests for quotations from suppliers.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Purchasing

Open Interface

Purchase Order Documents

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- PO_HEADERS_INTERFACE
- PO_LINES_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- PO_INTERFACE_ERRORS
- PO_HEADERS_INTERFACE

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
INTERFACE_TYPE	Error message source.
INTERFACE_TRANSACTION_ID	Error transaction unique ID.
COLUMN_NAME	Error column name.
ERROR_MESSAGE	Error message text.
PO_HEADER_ID	Unique purchase order header ID.
VENDOR_NAME	Vendor name.

Usage Comments

- You can import blanket purchase agreements or catalog quotations; you cannot import standard purchase orders.
- To import purchase order documents, run the Purchasing Documents Open Interface function, which is under the Oracle Purchasing module.

POReceiving

Use this transaction to import receipt information from other Oracle Applications or non-Oracle systems. You use this interface to integrate your Oracle purchasing application with new or existing applications.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Purchasing

Open Interface

Receiving

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- RCV_HEADERS_INTERFACE
- RCV_TRANSACTIONS_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- PO_INTERFACE_ERRORS
- RCV_HEADERS_INTERFACE
- RCV_TRANSACTIONS_INTERFACE

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
INTERFACE_TYPE	Error message source.
INTERFACE_TRANSACTION_ID	Error transaction unique ID.
COLUMN_NAME	Error column name.
ERROR_MESSAGE	Error message text.
PO_HEADER_ID	Unique purchase order header ID.
PO_LINE_ID	Unique purchase order line ID.
VENDOR_NAME	Vendor name.

Usage Comments

To import receipt information from open interface tables to internal tables, run the Receiving Transaction Processor function, which is under the Oracle Purchasing module.

PORequisitions

Use this transaction to import requisitions from other Oracle Applications or non-Oracle systems. This transaction enables you to integrate your Oracle Applications quickly with new or existing applications such as material replenishment, planning, inventory management, and production control systems.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Purchasing

Open Interface

PO Requisitions

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- PO_REQUISITIONS_INTERFACE
- PO_REQ_DIST_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- PO_INTERFACE_TABLES
- PO_RESCHEDULE_INTERFACE
- FND_USER_VIEW

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
INTERFACE_TYPE	Interface name; for example, Requisitions.
INTERFACE_TRANSACTION_ID	Transaction ID.
COLUMN_NAME	Column in which the error occurred.
ERROR_MESSAGE	Detailed explanation of the error.
SOURCE_TYPE_CODE	Source type code.
QUANTITY	Number of Items in the requisition.

Usage Comments

To import requisitions run the **Requisition Import** function, which is under the Oracle Purchasing module.

POResheduleRequisitions

Use this transaction to reschedule requisitions according to changes in your planned orders.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Purchasing

Open Interface

PO Reschedule Requisitions

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- PO_RESCHEDULE_INTERFACE

Error SQL

None.

Usage Comments

To import rescheduling information for requisitions, run the Reschedule Requisition function, which is under the Oracle Purchasing module.

CHAPTER 4

Transaction Definitions (Oracle Applications 11.0-to-IS)

Overview

This chapter provides an in-depth explanation of the Oracle Applications 11.0-to-IS sample transaction definitions. Oracle-to-IS transactions perform queries against the Oracle database and return the requested information to the client.

This chapter describes the transactions as they exist in the samples. It does not cover any modifications that you can make to the sample transactions.

You should already have an understanding of the webMethods Integration Server, the Oracle Applications Adapter, and the Oracle Application open interfaces.

For information about IS-to-Oracle Applications 11.0 transactions, see Chapter 3, "Transaction Definitions (IS-to-Oracle Applications 11.0)".

Transaction Overview

The sample transactions retrieve data from Oracle Applications. These transactions are triggered from within the webMethods Integration Server. A fully defined Oracle-to-IS transaction has two parts: a functional query and an acknowledgment update. The functional query has inputs into which the user maps query parameter values. The service then runs the query and the results are available as service outputs. The acknowledgment service updates the database to indicate that a record has been processed.

The current list of transactions includes the following:

- PurchaseOrderChange
- PurchaseOrderNew
- InvoiceOutbound

InvoiceOutbound

Use this transaction to retrieve new invoices in your Oracle Application system. You can retrieve invoices by creation date range, organization ID, billable customer name, the site use code, and transaction number. This transaction returns invoices that are completed and have the printing option turned on with a status of printing pending. The acknowledgement SQL marks the specified invoice so that the functional query will not retrieve it a second time.

Functional Query

Inputs

The inputs are used to limit which Invoices you want to retrieve. These SQL inputs are exposed as service inputs.

Outputs

Item Name	Description
ORG_ID	Self explanatory.
BILL_TO_CUSTOMER_NAME	Self explanatory.
SITE_USE_CODE	Self explanatory.
TRANSACTION_NUMBER	Self explanatory.
CREATION_DATE_FROM	Inclusive beginning date for CREATION_DATE field.
CREATION_DATE_TO	Inclusive ending date for CREATION_DATE field.

The output contains all invoices returned by the query. Each invoice consists of 1 header with 1 to N lines and each line contains 1 to N line tax records. The result will look like the following:

```
Header[0]
    Line[0]
        Line_Tax[0]
        …One Line_Tax record for each line tax
     …One line record each Invoice line item
… One Headers record for each Invoice
```

Header Details

The following table shows the details of an invoice header. The content of most fields can be determined by their name. Where necessary, a description is provided.

Item Name and Field Description (As Needed)
ORG_ID
DOCUMENT_TYPE
INVOICE_NAME
DOCUMENT_NUMBER
TRANSACTION_DATE
BILL_TO_CUSTOMER_CODE_INT
BILL_TO_CUSTOMER_NAME
BILL_TO_ADDRESS1
BILL_TO_ADDRESS2
BILL_TO_ADDRESS3

Item Name and Field Description (As Needed)
BILL_TO_ADDRESS4
BILL_TO_CITY
BILL_TO_POSTAL_CODE
BILL_TO_COUNTRY
BILL_TO_STATE
BILL_TO_PROVINCE
BILL_TO_COUNTY
BILL_TO_CUSTOMER_SIC_CODE
BILL_TO_CUSTOMER_SALES_CHANNEL
BILL_TO_CONTACT_LAST_NAME
BILL_TO_CONTACT_FIRST_NAME
BILL_TO_CONTACT_JOB_TITLE
BILL_TO_CUSTOMER_ATT_CATEGORY
BILL_TO_CUSTOMER_ATTRIBUTE1
BILL_TO_CUSTOMER_ATTRIBUTE2
BILL_TO_CUSTOMER_ATTRIBUTE3
BILL_TO_CUSTOMER_ATTRIBUTE4
BILL_TO_CUSTOMER_ATTRIBUTE5
BILL_TO_CUSTOMER_ATTRIBUTE6
BILL_TO_CUSTOMER_ATTRIBUTE7
BILL_TO_CUSTOMER_ATTRIBUTE8
BILL_TO_CUSTOMER_ATTRIBUTE9
BILL_TO_CUSTOMER_ATTRIBUTE10
BILL_TO_CUSTOMER_ATTRIBUTE11
BILL_TO_CUSTOMER_ATTRIBUTE12
BILL_TO_CUSTOMER_ATTRIBUTE13
BILL_TO_CUSTOMER_ATTRIBUTE14
BILL_TO_CUSTOMER_ATTRIBUTE15
SITE_USE_CODE
BILL_TO_SITE_ATT_CATEGORY

Item Name and Field Description (As Needed)
BILL_TO_SITE_ATTRIBUTE1
BILL_TO_SITE_ATTRIBUTE2
BILL_TO_SITE_ATTRIBUTE3
BILL_TO_SITE_ATTRIBUTE4
BILL_TO_SITE_ATTRIBUTE5
BILL_TO_SITE_ATTRIBUTE6
BILL_TO_SITE_ATTRIBUTE7
BILL_TO_SITE_ATTRIBUTE8
BILL_TO_SITE_ATTRIBUTE9
BILL_TO_SITE_ATTRIBUTE10
BILL_TO_SITE_ATTRIBUTE11
BILL_TO_SITE_ATTRIBUTE12
BILL_TO_SITE_ATTRIBUTE13
BILL_TO_SITE_ATTRIBUTE14
BILL_TO_SITE_ATTRIBUTE15
SHIP_TO_CUSTOMER_CODE_INT
SHIP_TO_CUSTOMER_NAME
SHIP_TO_ADDRESS1
SHIP_TO_ADDRESS2
SHIP_TO_ADDRESS3
SHIP_TO_ADDRESS4
SHIP_TO_CITY
SHIP_TO_POSTAL_CODE
SHIP_TO_COUNTRY
SHIP_TO_STATE
SHIP_TO_PROVINCE
SHIP_TO_COUNTY
SHIP_TO_CUSTOMER_SIC_CODE
SHIP_TO_CUSTOMER_SALES_CHANNEL
SHIP_TO_CONTACT_LAST_NAME

Item Name and Field Description (As Needed)
SHIP_TO_CONTACT_FIRST_NAME
SHIP_TO_CONTACT_JOB_TITLE
SOLD_TO_CUSTOMER_CODE_INT
SOLD_TO_CUSTOMER_NAME
SOLD_TO_ADDRESS1
SOLD_TO_ADDRESS2
SOLD_TO_ADDRESS3
SOLD_TO_ADDRESS4
SOLD_TO_CITY
SOLD_TO_POSTAL_CODE
SOLD_TO_COUNTRY
SOLD_TO_STATE
SOLD_TO_PROVINCE
SOLD_TO_COUNTY
SOLD_TO_CUSTOMER_SIC_CODE
SOLD_TO_CUSTOMER_SALES_CHANNEL
SOLD_TO_CONTACT_LAST_NAME
SOLD_TO_CONTACT_FIRST_NAME
SOLD_TO_CONTACT_JOB_TITLE
TRANSACTION_NUMBER
CREDITED_INVOICE_NUMBER
REFERENCE_INVOICE_NUMBER
PARENT_INVOICE_NUMBER
SHIPMENT_DATE
PURCHASE_ORDER_NUMBER
CREATION_DATE
PURCHASE_ORDER_REVISION_NUMBER
COMMITMENT_START_DATE
PURCHASE_ORDER_DATE
COMMITMENT_END_DATE

Item Name and Field Description (As Needed)
INV_TRANSACTION_DATE
LAST_UPDATE_DATE
DUE_DATE
CREDIT_MEMO_REASON
TRANSMISSION_LEAD_DAYS
TRANSACTION_SOURCE
INSTALLMENT_NUMBER
SHIPMENT_WAYBILL_NUMBER
SHIP_VIA
SHIPMENT_FOB_POINT
CURRENCY_CODE
CURRENCY_EXCHANGE_RATE
BASE_CURRENCY_CODE
PAYMENT_TERM_NAME
PRIMARY_SALESREP_NAME
COMMENTS
BILL_TO_ADDRESS_ID
BILL_TO_CUSTOMER_LOCATION
BILL_TO_CUSTOMER_NUMBER
SHIP_TO_ADDRESS_ID
SHIP_TO_CUSTOMER_LOCATION
SHIP_TO_CUSTOMER_NUMBER
SOLD_TO_ADDRESS_ID
SOLD_TO_CUSTOMER_LOCATION
SOLD_TO_CUSTOMER_NUMBER
TRANSACTION_ID
PAYMENT_TERM_ID
BILL_TO_CUSTOMER_ID
BILL_TO_SITE_USE_ID
SHIP_TO_CUSTOMER_ID

Item Name and Field Description (As Needed)
SHIP_TO_SITE_USE_ID
SOLD_TO_CUSTOMER_ID
SOLD_TO_SITE_USE_ID
TERM_DUE_CUTOFF_DAY
TERM_DUE_DATE
TERM_DUE_DAYS
TERM_DUE_DAY_OF_MONTH
TERM_DUE_MONTHS_FORWARD
TERM_DUE_PERCENT
REMIT_TO_ADDRESS1
REMIT_TO_ADDRESS2
REMIT_TO_ADDRESS3
REMIT_TO_ADDRESS4
REMIT_TO_CITY
REMIT_TO_COUNTY
REMIT_TO_STATE
REMIT_TO_PROVINCE
REMIT_TO_COUNTRY
REMIT_TO_CODE_INT
REMIT_TO_POSTAL_CODE
MULTIPLE_INSTALLMENTS_FLAG
MAXIMUM_INSTALLMENT_NUMBER
AMOUNT_TAX_DUE
AMOUNT_CHARGES_DUE
AMOUNT_FREIGHT_DUE
AMOUNT_LINE_INVITEMS_DUE
TOTAL_AMOUNT_DUE
DISCOUNT_PERCENT1
DISCOUNT_DAYS1
DISCOUNT_DATE1

Item Name and Field Description (As Needed)
DISCOUNT_DAY_OF_MONTH1
DISCOUNT_MONTHS_FORWARD1
DISCOUNT_PERCENT2
DISCOUNT_DAYS2
DISCOUNT_DATE2
DISCOUNT_DAY_OF_MONTH2
DISCOUNT_MONTHS_FORWARD2
DISCOUNT_PERCENT3
DISCOUNT_DAYS3
DISCOUNT_DATE3
DISCOUNT_DAY_OF_MONTH3
DISCOUNT_MONTHS_FORWARD3

Line Details

The following table shows the details of an invoice line item. The content of most fields can be determined by their names. Where necessary, a description is provided.

Item Name and Field Description (as needed)
ORG_ID
TRANSACTION_ID
LINE_NUMBER
SALES_ORDER_NUMBER
SALES_ORDER_REVISION_NUMBER
SALES_ORDER_LINE_NUMBER
SALES_ORDER_DATE
SALES_CHANNEL
ITEM_ID
CUSTOMER_ITEM_NUMBER
CUSTOMER_ITEM_DESC
ITEM_DESCRIPTION
UOM_CODE

Item Name and Field Description (as needed)
ORDERED_QUANTITY
QUANTITY
UNIT_STANDARD_PRICE
UNIT_SELLING_PRICE
LINE_AMOUNT
CREDIT_MEMO_REASON
CREDITED_LINE_NUMBER
SHIP_ORDER_STATUS_INT
TRANSACTION_REFERENCE_KEY
INTERFACE_LINE_CATEGORY
INTERFACE_LINE_ATTRIBUTE1
INTERFACE_LINE_ATTRIBUTE2
INTERFACE_LINE_ATTRIBUTE3
INTERFACE_LINE_ATTRIBUTE4
INTERFACE_LINE_ATTRIBUTE5
INTERFACE_LINE_ATTRIBUTE6
INTERFACE_LINE_ATTRIBUTE7
INTERFACE_LINE_ATTRIBUTE8
INTERFACE_LINE_ATTRIBUTE9
INTERFACE_LINE_ATTRIBUTE10
INTERFACE_LINE_ATTRIBUTE11
INTERFACE_LINE_ATTRIBUTE12
INTERFACE_LINE_ATTRIBUTE13
INTERFACE_LINE_ATTRIBUTE14
INTERFACE_LINE_ATTRIBUTE15
LINE_ATTRIBUTE_CATEGORY
LINE_ATTRIBUTE1
LINE_ATTRIBUTE2
LINE_ATTRIBUTE3
LINE_ATTRIBUTE4

Item Name and Field Description (as needed)
LINE_ATTRIBUTE5
LINE_ATTRIBUTE6
LINE_ATTRIBUTE7
LINE_ATTRIBUTE8
LINE_ATTRIBUTE9
LINE_ATTRIBUTE10
LINE_ATTRIBUTE11
LINE_ATTRIBUTE12
LINE_ATTRIBUTE13
LINE_ATTRIBUTE14
LINE_ATTRIBUTE15

Line_Tax Details

The following table shows the details of an invoice line item tax record. The content of most fields can be determined by their names. Where necessary, a description is provided.

Item Name and Field Description (as needed)
ORG_ID
TRANSACTION_ID
LINE_NUMBER
CUSTOMER_TRX_LINE_ID
TAX_LINE_NUMBER
LINE_TYPE
TAX_AMOUNT
TAX_RATE
TAX_PRECEDENCE
TAX_CODE
LINK_TO_CUST_TRX_LINE_ID
TAX_ATTRIBUTE_CATEGORY
TAX_ATTRIBUTE1
TAX_ATTRIBUTE2

Item Name and Field Description (as needed)
TAX_ATTRIBUTE3
TAX_ATTRIBUTE4
TAX_ATTRIBUTE5
TAX_ATTRIBUTE6
TAX_ATTRIBUTE7
TAX_ATTRIBUTE8
TAX_ATTRIBUTE9
TAX_ATTRIBUTE10
TAX_ATTRIBUTE11
TAX_ATTRIBUTE12
TAX_ATTRIBUTE13
TAX_ATTRIBUTE14
TAX_ATTRIBUTE15
TAX_EXEMPT_FLAG
TAX_EXEMPT_NUMBER
TAX_EXEMPT_REASON_CODE
TAX_TYPE
DESCRIPTION
LOCATION
TAX_CLASSIFICATION
VAT_TRANSACTION_TYPE
VAT_TAX_ATTRIBUTE_CATEGORY
VAT_TAX_ATTRIBUTE1
VAT_TAX_ATTRIBUTE2
VAT_TAX_ATTRIBUTE3
VAT_TAX_ATTRIBUTE4
VAT_TAX_ATTRIBUTE5
VAT_TAX_ATTRIBUTE6
VAT_TAX_ATTRIBUTE7
VAT_TAX_ATTRIBUTE8

Item Name and Field Description (as needed)
VAT_TAX_ATTRIBUTE9
VAT_TAX_ATTRIBUTE10
VAT_TAX_ATTRIBUTE11
VAT_TAX_ATTRIBUTE12
VAT_TAX_ATTRIBUTE13
VAT_TAX_ATTRIBUTE14
VAT_TAX_ATTRIBUTE15

Acknowledgment SQL

Updates the specified invoice to show it has been processed and printed.

Inputs

The inputs define which PO to update.

- DOCUMENT_TYPE
- TRANSACTION_ID
- INSTALLMENT_NUMBER
- MULTIPLE_INSTALLMENTS_FLAG
- MAXIMUM_INSTALLMENT_NUMBER

Outputs

There is no output generated by this SQL.

Usage Comments

Invoices are retrieved using the `TRANSACTION_NUMBER` field, but they are acknowledged using the `TRANSACTION_ID` field.

PurchaseOrderChange

Use this transaction to retrieve revised purchase orders. You can retrieve new purchase orders by creation date range, supplier, vendor, PO type, and PO number range. This transaction only retrieves Purchase Orders that fit the above criteria and have already been printed or marked as processed. The acknowledgement SQL marks the specified PO so that it will not be retrieved by the functional query a second time.

Functional Query

Inputs

Same as the PurchaseOrderNew transaction.

Outputs

Same as the PurchaseOrderNew transaction, except for the differences noted below.

Header Details

Does not have a `TRANSACTION_DATE` field.

Line Details

Does not have an `ORG_ID` field.

Shipment Details

Does not have an `ORG_ID` field.

Contains a `QUANTITY_PENDING` field

Acknowledgment SQL

Same as for the PurchaseOrderNew transaction.

Inputs

Same as for the PurchaseOrderNew transaction.

Outputs

Same as for the PurchaseOrderNew transaction.

Usage Comments

- Purchase Orders that have been approved and previously printed are retrieved.
- You can retrieve only BLANKET and STANDARD Purchase Orders Types. To retrieve a STANDARD purchase order, specify a DOCUMENT_TYPE of STANDARD. To retrieve the purchase agreements, specify a DOCUMENT_TYPE of BLANKET. To retrieve releases against a purchase agreement, specify a DOCUMENT_TYPE of RELEASE.

PurchaseOrderNew

Use this transaction to retrieve new purchase orders entered into your system. You can retrieve new purchases by creation date range, supplier, vendor, PO type, and PO number range. This transaction only retrieves Purchase Orders that fit the above criteria and have not been marked as printed. The acknowledgement SQL marks the specified PO as printed so that the functional query will not retrieve it a second time.

Functional Query

Inputs

The inputs are used to limit what Purchase Orders you want to retrieve. These SQL inputs are exposed as service inputs.

Item Name	Description
CREATE_DATE_FROM	Inclusive beginning date for PO CREATION_DATE field.
CREATE_DATE_TO	Inclusive ending date for PO CREATION_DATE field.
SUPPLIER_NUMBER	Used to limit new purchase orders retrieved to this supplier.
VENDOR_SITE_ID	Used to limit new purchase orders retrieved to this vendor site.
DOCUMENT_TYPE	Used to limit new purchase orders retrieved to this document type (STANDARD, BLANKET).
PO_NUMBER_FROM	Inclusive beginning PO number.
PO_NUMBER_TO	Inclusive ending PO number.

Outputs

The output contains all purchase orders returned by the query. Each purchase order consists of 1 header with 1 to N lines and each line contains 1 to N shipment records. The result looks like the following:

```
Headers[0]
    Lines[0]
        Shipments[0]
        …One Shipments record for each shipment record
     …One Lines record each line item
… One Headers record for each Purchase Order
```

Header Details

A header consists of the following fields. The content of most fields can be determined by their name. Where necessary, a description is provided.

Item Name and Field Description (as needed)
ORG_ID
DOCUMENT_TYPE Description: STANDARD, BLANKET, or RELEASE.
DOCUMENT_CODE
TRANSACTION_DATE
PO_NUMBER
POR_RELEASE_ID Description: Purchase order release information.
POR_RELEASE_NUM
POR_RELEASE_DATE
CREATION_DATE
REVISION_NUM
REVISED_DATE
COMMENTS
PO_TYPE Description: STANDARD or BLANKET.
PAYMENT_TERMS
CURRENCY_CODE

Item Name and Field Description (as needed)
CURRENCY_RATE
SHIP_VIA
FOB_CODE
FREIGHT_TERMS
CANCEL_FLAG
ACCEPTANCE_REQUIRED_FLAG
ACCEPTANCE_DUE_DATE
CONFIRMING_ORDER_FLAG
BLANKET_START_DATE
BLANKET_END_DATE
BLANKET_TOTAL_AMOUNT
SUPPLIER_NUMBER
SUPPLIER_NAME
NOTE_TO_VENDOR
PO_ATTRIBUTE_CATEGORY Description: Purchase order attributes.
PO_ATTRIBUTE1
PO_ATTRIBUTE2
PO_ATTRIBUTE3
PO_ATTRIBUTE4
PO_ATTRIBUTE5
PO_ATTRIBUTE6
PO_ATTRIBUTE7
PO_ATTRIBUTE8
PO_ATTRIBUTE9
PO_ATTRIBUTE10
PO_ATTRIBUTE11
PO_ATTRIBUTE12
PO_ATTRIBUTE13
PO_ATTRIBUTE14

Item Name and Field Description (as needed)
PO_ATTRIBUTE15
SU_ATTRIBUTE_CATEGORY Description: Supplier attribute information.
SU_ATTRIBUTE1
SU_ATTRIBUTE2
SU_ATTRIBUTE3
SU_ATTRIBUTE4
SU_ATTRIBUTE5
SU_ATTRIBUTE6
SU_ATTRIBUTE7
SU_ATTRIBUTE8
SU_ATTRIBUTE9
SU_ATTRIBUTE10
SU_ATTRIBUTE11
SU_ATTRIBUTE12
SU_ATTRIBUTE13
SU_ATTRIBUTE14
SU_ATTRIBUTE15
SS_ATTRIBUTE_CATEGORY Description: Supplier site attributes.
SS_ATTRIBUTE1
SS_ATTRIBUTE2
SS_ATTRIBUTE3
SS_ATTRIBUTE4
SS_ATTRIBUTE5
SS_ATTRIBUTE6
SS_ATTRIBUTE7
SS_ATTRIBUTE8
SS_ATTRIBUTE9
SS_ATTRIBUTE10

Item Name and Field Description (as needed)
SS_ATTRIBUTE11
SS_ATTRIBUTE12
SS_ATTRIBUTE13
SS_ATTRIBUTE14
SS_ATTRIBUTE15
CUSTOMER_NUMBER Description: Vendor customer number.
VENDOR_SITE_ID
SU_ADDRESS_LINE1
SU_ADDRESS_LINE2
SU_ADDRESS_LINE3
SU_ADDRESS_LINE4
SU_CITY
SU_ZIP
SU_COUNTRY
SU_STATE
SU_PROVINCE
SU_AREA_CODE
SU_PHONE
SU_FAX_AREA_CODE
SU_FAX
SU_TELEX
CN_LAST_NAME Description: Vendor contact information.
CN_FIRST_NAME
CN_AREA_CODE
CN_PHONE
ST_LOCATION_ID Description: Ship-to information.
ST_CONTACT_LAST_NAME

Item Name and Field Description (as needed)
ST_CONTACT_FIRST_NAME
ST_NAME
ST_ADDRESS_LINE1
ST_ADDRESS_LINE2
ST_ADDRESS_LINE3
ST_CITY
ST_POSTAL_CODE
ST_COUNTRY
ST_REGION1
ST_REGION2
ST_REGION3
ST_PHONE1
ST_PHONE2
ST_PHONE3
BT_LOCATION_ID Description: Bill-to information.
BT_CONTACT_LAST_NAME
BT_CONTACT_FIRST_NAME
BT_NAME
BT_ADDRESS_LINE1
BT_ADDRESS_LINE2
BT_ADDRESS_LINE3
BT_CITY
BT_POSTAL_CODE
BT_COUNTRY
BT_REGION1
BT_REGION2
BT_REGION3
BT_PHONE1
BT_PHONE2

Item Name and Field Description (as needed)
BT_PHONE3
BUYER_LAST_NAME
BUYER_FIRST_NAME
BUYER_EMAIL_ADDRESS
BUYER_WORK_TELEPHONE
BUYER_MISC_TELEPHONE1
BUYER_MISC_TELEPHONE2
BUYER_MISC_TELEPHONE3
PO_HEADER_ID
FSP_INVENTORY_ORGANIZATION_ID Description: Financial system parameter inventory organization ID.

Line Details

Line Item details consist of the following fields:

Item Name and Description (as needed)
ORG_ID
LINE_NUM
QUANTITY
UOM_CODE Description: Unit of measure code.
ITEM_ID Description: Inventory item ID.
ITEM_REVISION
VENDOR_PRODUCT_NUMBER
UNIT_PRICE
ITEM_DESCRIPTION
PO_NUMBER
VENDOR_QUOTE_NUMBER
CANCEL_FLAG
CANCEL_DATE

Item Name and Description (as needed)
QUANTITY_COMMITTED
COMMITTED_AMOUNT
LIST_PRICE_PER_UNIT
MARKET_PRICE
NOT_TO_EXCEED_PRICE
NEGOTIATED_BY_PREPARER_FLAG
TAXABLE_FLAG
TRANSACTION_REASON_CODE
LINE_TYPE
HAZARD_CLASS
UN_NUMBER
UN_DESCRIPTION
NOTE_TO_VENDOR
POL_ATTRIBUTE_CATEGORY Description: Purchase order line attributes.
POL_ATTRIBUTE1
POL_ATTRIBUTE2
POL_ATTRIBUTE3
POL_ATTRIBUTE4
POL_ATTRIBUTE5
POL_ATTRIBUTE6
POL_ATTRIBUTE7
POL_ATTRIBUTE8
POL_ATTRIBUTE9
POL_ATTRIBUTE10
POL_ATTRIBUTE11
POL_ATTRIBUTE12
POL_ATTRIBUTE13
POL_ATTRIBUTE14
POL_ATTRIBUTE15

Item Name and Description (as needed)
MTL_ORGANIZATION_ID
LP_ATTRIBUTE_CATEGORY Description: Material item attributes.
LP_ATTRIBUTE1
LP_ATTRIBUTE2
LP_ATTRIBUTE3
LP_ATTRIBUTE4
LP_ATTRIBUTE5
LP_ATTRIBUTE6
LP_ATTRIBUTE7
LP_ATTRIBUTE8
LP_ATTRIBUTE9
LP_ATTRIBUTE10
LP_ATTRIBUTE11
LP_ATTRIBUTE12
LP_ATTRIBUTE13
LP_ATTRIBUTE14
LP_ATTRIBUTE15
LP_SEGMENT1
LP_SEGMENT2
LP_SEGMENT3
LP_SEGMENT4
LP_SEGMENT5
LP_SEGMENT6
LP_SEGMENT7
LP_SEGMENT8
LP_SEGMENT9
LP_SEGMENT10
LP_SEGMENT11
LP_SEGMENT12

Item Name and Description (as needed)
LP_SEGMENT13
LP_SEGMENT14
LP_SEGMENT15
LP_SEGMENT16
LP_SEGMENT17
LP_SEGMENT18
LP_SEGMENT19
LP_SEGMENT20
POR_RELEASE_ID
POR_RELEASE_NUM
POR_SOURCE_LINE_NUM
PO_HEADER_ID Description: Shows relationship to PO header.
PO_LINE_ID
SHIPMENT_TYPE
SHIPMENTS Description: Start of shipment record.

Shipment Details

Shipment details consist of the following fields:

Item Name	Description
ORG_ID	
SHIPMENT_NUMBER	
QUANTITY_ORIGINAL	
UOM_CODE Description: Unit of measure code.	Unit of measure code.
SHIPMENT_NEED_BY_DATE	
SHIPMENT_PROMISED_DATE	
SHIPMENT_LAST_ACCEPTABLE_DATE	
QUANTITY_CANCELLED	

Item Name	Description
QUANTITY_RECEIVED	
PRICE_OVERRIDE	
CANCELLED_FLAG	
CANCELLED_DATE	
SHIP_VIA	
FOB_CODE	
FREIGHT_TERMS	
TAXABLE_FLAG	
SHIPMENT_ATTRIBUTE_CATEGORY	
SHIPMENT_ATTRIBUTE1	
SHIPMENT_ATTRIBUTE2	
SHIPMENT_ATTRIBUTE3	
SHIPMENT_ATTRIBUTE4	
SHIPMENT_ATTRIBUTE5	
SHIPMENT_ATTRIBUTE6	
SHIPMENT_ATTRIBUTE7	
SHIPMENT_ATTRIBUTE8	
SHIPMENT_ATTRIBUTE9	
SHIPMENT_ATTRIBUTE10	
SHIPMENT_ATTRIBUTE11	
SHIPMENT_ATTRIBUTE12	
SHIPMENT_ATTRIBUTE13	
SHIPMENT_ATTRIBUTE14	
SHIPMENT_ATTRIBUTE15	
SHIP_TO_LOCATION_ID	
SHIP_TO_LOCATION_CODE	
SHIP_TO_CONTACT_LAST_NAME	
SHIP_TO_CONTACT_FIRST_NAME	
SHIP_TO_ADDRESS_LINE_1	
SHIP_TO_ADDRESS_LINE_2	

Item Name	Description
SHIP_TO_ADDRESS_LINE_3	
SHIP_TO_CITY	
SHIP_TO_POSTAL_CODE	
SHIP_TO_COUNTRY	
SHIP_TO_REGION_1	
SHIP_TO_REGION_2	
SHIP_TO_REGION_3	
SHIP_TO_PHONE1	
SHIP_TO_PHONE2	
SHIP_TO_PHONE3	
PO_HEADER_ID Description: Shows relationship to PO header.	
PO_LINE_ID Description: Shows relationship to PO header.	
POR_RELEASE_ID	
POR_RELEASE_NUM	
LINE_LOCATION_ID	

Acknowledgment SQL

Updates the specified purchase order to show it has been processed. It marks the PO as printed and updates the print date, edi_processed_flag, and the last update date. It also performs the necessary archive procedures. It updates and archives a single PO at a time.

Inputs

The inputs define which PO to update.

- ORG_ID
- DOCUMENT_TYPE
- PO_NUMBER
- PO_TYPE
- POR_RELEASE_NUMBER

Outputs

There is no output generated by this SQL.

Usage Comments

- Purchase Orders that have been approved and not printed are retrieved.
- Purchase Orders that are Cancelled or on Hold are not retrieved.
- Only BLANKET and STANDARD Purchase Orders Types can be retrieved.

webMethods.

CHAPTER 5

Transaction Definitions (IS-to-Oracle Applications 10.7SC)

Overview

This chapter provides an in-depth explanation of the IS-to-Oracle Applications 10.7SC sample transaction definitions. IS-to-Oracle Applications transactions update the Oracle database with information from an end-user or XML file.

This chapter describes the transactions as they exist in the samples. It does not cover any modifications that you can make to the sample transactions.

You should already have an understanding of the webMethods Integration Server, the Oracle Application Adapter, and the Oracle Application open interfaces.

For information about Oracle Applications 10.7SC-to-IS transactions, see Chapter 6, "Transaction Definitions (Oracle Applications 10.7SC-to-IS)".

Transaction Overview

The sample transactions insert data into Oracle Applications open interface tables. A fully defined IS-to-Oracle transaction has two parts: the open interface table(s) and an error SQL query. The transaction may group one or more open interface tables together. Every column from each open interface table for the defined transaction will be exposed as input to the adapter service. After the service inserts records into corresponding open interface tables, Oracle Applications must run the respective import process. When completed, you can test the transaction for errors from the webMethods Integration Server by using the error SQL query. This has inputs into which the user maps query parameter values. The service then runs the query and the results are available as service outputs.

For more information about the IS-to-Oracle Applications 10.7SC transactions, see your Oracle Applications documentation.

This chapter shows the transactions in alphabetical order. The table below shows the transactions organized by Oracle Applications module.

Module	webMethods Transaction	Page
Payables	"APInvoice"	99
Receivables	"ARCustomers"	100
Inventory	"INVDemand"	102
	"INVItems"	104
	"INVOpenTransactions"	106
	"INVReplenishment"	107

Module	webMethods Transaction	Page
Order Entry	“OEOrderImport”	109
	“OEShipConfirm”	111
Purchasing	“PORequisitions”	112
	“POReScheduleRequisitions”	114

APInvoice

Use this transaction to create invoices from invoice information loaded from other accounting system and to update voucher number information for existing invoices. The Payables Open Interface Import program creates invoices to import into Payables using invoice data stored in the Payables open interface.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Payables

Open Interface

AP Invoice Open Import Transaction

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- AP_EXPENSE_REPORT_HEADERS_ALL
- AP_EXPENSE_REPORT_LINES_ALL

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- AP_EXPENSE_REPORT_HEADERS_ALL
- FND_USER

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.
SOURCE	Import source name for your non-Oracle accounting system.

Outputs

Usage Comments

Field	Description
VENDOR_ID	Vendor ID.
VENDOR_SITE_ID	Vendor site ID.
INVOICE_NUM	Invoice number.
SOURCE	Invoice ID.
REJECT_CODE	Description.

To import invoice information, run the Payables Invoice Import request, which is under the AP module.

ARCustomers

Use this transaction to create and import customers, customer profiles, addresses, contacts, and phone numbers from other systems.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Accounts Receivable

Open Interface

Customer Transaction

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- RA_CUSTOMERS_INTERFACE_ALL
- RA_CUSTOMERS_BANKS_INT_ALL
- RA_CUST_PAY_METHOD_INT_ALL
- RA_CONTACT_PHONES_INT_ALL
- RA_CUSTOMER_PROFILES_INT_ALL

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- RA_CUSTOMERS_INTERFACE_ALL
- FND_USER

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.
CUSTOMER_NAME	Customer name to be imported.

Outputs

Field	Description
ORIG_SYSTEM_CUSTOMER_REF	Original system customer reference (i.e. primary key for customer in other application).
CUSTOMER_NAME	Name of the customer to be imported.
INTERFACE_STATUS	Error codes. For error code definitions, see http://www.oracle.com/.
PROCESS_MODE	Process flag to indicate whether processing of row succeeded or failed.
CUSTOMER_NAME	Customer name.
ERROR_EXPLANATION	Detailed error message.

Usage Comments

- To import invoice information, run the Customer Interface request, which is under the AR module.
- You can run Customer Interface Transfer report to see exact errors .
- INACTIVE_FLAG is a required field. Set it to “1” for Yes and “2” for No.

INVDemand

Use this transaction to integrate an external Order Entry with an Oracle Applications Inventory. You can add a demand, add and modify a reservation, query the on-hand quantity, change available quantities to reserve quantities, change available quantities to promised quantities, and reserve on-hand quantities.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Inventory

Open Interface

Demand Transaction

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- MTL_DEMAND_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- MTL_DEMAND_INTERFACE
- MFG_LOOKUPS
- FND_USER

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
SCHEDULE_GROUP_ID	Group ID.
DEMAND_SOURCE_TYPE	Origination of the demand or reservation (e.g. 2-Sales Order).
ACTION_CODE	Code used to identify what action has to be taken with this record.
ORGANIZATION_ID	Organization ID.
INVENTORY_ITEM_ID	Item ID.
ERROR	Error message.
USER_NAME	User name.

Usage Comments

- To import demand information, run the Process Demand Interface request, which is under the Background module.
- In MTL_DEMAND_INTERFACE there are cases when an error_code is generated that does not provide an error_explanation.

INVItems

Use this transaction to convert inventory Items from another inventory system, migrate assembly and component items from a legacy manufacturing system, convert purchased items from a custom purchasing system, and insert new items from a Product Data Management package. When importing items, this transaction creates new Items in the Item Master organization or assigns existing Items to additional organizations. You can specify values for all Item attributes, or you can use default or null values. The INVItems transaction also enables you to import revision details, including past and future revisions and effective dates.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Inventory

Open Interface

Items Transaction

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- MTL_SYSTEM_ITEM_INTERFACE
- MTL_ITEM_REVISIONS_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- MTL_SYSTEM_ITEM_INTERFACE
- MTL_INTERFACE_ERRORS
- FND_USER_VIEW

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
ITEM_NUMBER	Item number in old system.
DESCRIPTION	Description of item.
TRANSACTION_TYPE	Transaction type (CREATE or UPDATE).
PROCESS_FLAG	Process flag to indicate whether processing of row has succeeded or failed. Possible process flags are: 1 Pending 2 Assign complete 3 Assign/validation failed 4 Validation succeeded; import failed 5 Import in process 6 Import succeeded
ERROR_MESSAGE	Detailed error message.

Usage Comments

- To create Items, choose `TRANSACTION TYPE` as `CREATE`. When importing these Items, choose **Create Items**.
- To update Items, choose `TRANSACTION_TYPE` as `UPDATE`. When importing these Items choose **Update existing Items**.
- If you have a list of transactions containing both UPDATES and CREATES, give all CREATES one `SET_PROCESS_ID` and all `UPDATES` a different `SET_PROCESS_ID`. When importing you specify Items to import by their `SET_PROCESS_ID`.
- Be careful when running the import process. We have experienced a problem running Oracle Applications 10.7SC on Windows NT. If you specify a `SET_PROCESS_ID` used for your CREATE records and then during the import you specify the `SET_PROCESS_ID` for UPDATE records, a Dr. Watson Error will be triggered on

Oracle Applications. The process flag on the records will show pending and the `SET_PROCESS_ID` will be set to –999.

- Oracle Applications does not allow the deletion of Items using the open interface tables. CREATE and UPDATE transaction types are currently supported.
- To import Items, run the Import Items function, which is under the Inventory module.

INVOpenTransactions

Use this transaction to load transactions from external applications and feeder systems, which include sales, order shipment transactions from an order entry system other than Oracle Order Entry, simple material issues, receipts, or transfers.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Inventory

Open Interface

Inventory Open Transaction

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- MTL_TRANSACTIONS_INTERFACE
- MTL_TRANSACTION_LOTS_INTERFACE
- MTL_SERIAL_NUMBERS_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- MTL_TRANSACTIONS_INTERFACE
- FND_USER_VIEW

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
INVENTORY_ITEM_ID	Inventory item ID.
TRANSACTION_QUANTITY	Number of Items in the transaction.
ERROR_EXPLANATION	Detailed error explanation.
TRANSACTION_SOURCE_NAME	Name of the source.

Usage Comments

To import transactions, run the Import Open Transactions function, which is under the Inventory module.

INVReplenishment

Use this transaction to load replenishment request (stock take count, requisition request for non-trackable sub-inventory) from external systems.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Inventory

Open Interface

Replenishment Open Transaction

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- MTL_REPLENISH_HEADERS_INT
- MTL_REPLENISH_LINES_INT

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- MTL_REPLENISH_HEADERS_INT
- MTL_REPLENISH_LINES_INT
- MFG_LOOKUPS
- FND_USER

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.
COUNT_NAME	

Outputs

Field	Description
INVENTORY_ITEM_ID	Inventory item ID.
TRANSACTION_QUANTITY	Number of Items in the transaction.
ERROR_EXPLANATION	Detailed error explanation.
TRANSACTION_SOURCE_NAME	Name of the source.

Usage Comments

To import transactions, run the Import Open Transactions function, which is under the Inventory module.

OEOrderImport

Use this transaction to import order entry data from a variety of sources. You can import data into orders with lines, schedule details, price adjustments, and sales orders. You can import complete, incomplete, and booked orders. You can also import quota, or non-quota sales credit information, changes to existing orders, line schedule detail information with each order and order line, and internal requisition orders from Oracle Purchasing.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Order Entry/Shipping

Open Interface

Order Import

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- SO_HEADERS_INTERFACE_ALL
- SO_LINES_INTERFACE_ALL
- SO_LINES_DETAIL_INTERFACE
- SO_SALES_CREDITS_INTERFACE
- SO_PRICE_ADJUSTMENTS_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- SO_HEADERS_INTERFACE_ALL
- FND_USER_VIEW

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.
ORIG_SYSTEM_REF	Original system reference.
PO_NUMBER	Purchase order number.
CUSTOMER_NAME	Customer name.
CUSTOMER_NUMBER	Customer number in legacy system.

Outputs

Field	Description
ORIGINAL_SYSTEM_REFERENCE	Original system reference.
PURCHASE_ORDER_NUM	Purchase order number.
CUSTOMER_NUMBER	Customer number in legacy system.
CUSTOMER_NAME	Customer name.
INTERFACE_STATUS	Current status of the transaction (pending, rejected, or success).

Usage Comments

- For each record in the Line Interface, a corresponding record should exist in the Headers Table. Otherwise, the Lines record will be left dangling and will not be picked up by Import Process.
- You can enter Header Information without having Lines Detail.
- To import Items, run the Import Sales Order function, which is under the Order Entry module.

OEShipConfirm

Use this transaction to import shipping information into pick tables. You can also use this transaction to close the pick slip without explicitly navigating to the Confirm Shipments screen.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Order Entry/Shipping

Open Interface

Ship Confirm Import

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- WSH_PICKING_HEADERS_INTERFACE
- WSH_PICKING_DETAILS_INTERFACE
- WSH_FREIGHT_CHARGES_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- WSH_PICKING_HEADERS_INTERFACE
- FND_USER

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.
PICK_SLIP_NUMBER	Pick slip number.

Outputs

Field	Description
TRANSACTION_ID	Transaction ID.
PICK_SLIP_NUMBER	Pick slip number.
WAY_BILL_NUMBER	Waybill number.
DATE_SHIPPED	Shipping date.
ACTION_CODE	Action to be taken.
ERROR_EXPLANATION	Error message.

Usage Comments

To import shipping information, run the Ship Confirm Open Interface function, which is under the Order Entry/Confirm Shipping module.

PORequisitions

Use this transaction to import requisitions from other Oracle Applications or non-Oracle systems. This transaction enables you to integrate your Oracle Applications quickly with new or existing applications, such as material replenishment, planning, inventory management, and production control systems.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Purchasing

Open Interface

Requisitions

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- PO_REQUISITIONS_INTERFACE

Error SQL

Open interface table(s) queried by the error SQL are as follows:

- PO_INTERFACE_ERRORS
- PO_REQUISITIONS_INTERFACE
- FND_USER_VIEW

Inputs

Field	Description
WM_USER	User ID of the person who initiated the transaction; that is, the user ID in the CREATED_BY field in the transaction.

Outputs

Field	Description
INTERFACE_TYPE	Interface name; for example, Requisitions.
INTERFACE_TRANSACTION_ID	Transaction ID.
COLUMN_NAME	Column in which the error occurred.
ERROR_MESSAGE	Detailed explanation of the error.
SOURCE_TYPE_CODE	Source type code.
QUANTITY	Number of Items in the requisition.

Usage Comments

To import requisitions run the Requisition Import Request function, which is under the Oracle Purchasing module.

PORescheduleRequisitions

Use this transaction to reschedule requisitions according to changes in your planned orders.

For more information about this transaction, see your Oracle Applications documentation.

Oracle Applications Module

Purchasing

Open Interface

Requisitions

Open Interface Tables

Open interface table(s) exposed by the transaction are as follows:

- PO_RESCHEDULE_INTERFACE

Error SQL

None.

Usage Comments

To import rescheduling information for requisitions, run the Reschedule Requisition Request function, which is under the Oracle Purchasing module.

webMethods.

CHAPTER 6

Transaction Definitions (Oracle Applications 10.7SC-to-IS)

Overview

This chapter provides an in-depth explanation of the Oracle Applications 10.7SC-to-IS sample transaction definitions. Oracle Applications-to-IS transactions perform queries against the Oracle database and return the requested information to the client.

This chapter describes the transactions as they exist in the samples. It does not cover any modifications that you can make to the sample transactions.

You should already have an understanding of the webMethods Integration Server, the Oracle Applications Adapter, and the Oracle Application open interfaces.

For information about IS-to-Oracle Applications 10.7SC transactions, see Chapter 5, "Transaction Definitions (IS-to-Oracle Applications 10.7SC)".

Transaction Overview

The sample transactions retrieve data from Oracle Applications. These transactions are triggered from within the webMethods Integration Server. A fully defined Oracle-to-IS transaction has two parts: a functional query and an acknowledgment update. The functional query will have inputs into which the user maps query parameter values. The service then runs the query and the results are available as service outputs. The acknowledgment service updates the database to indicate that a record has been processed.

The current list of transactions includes the following:

- AdvanceShipNotice
- InvoiceOutbound
- PurchaseOrderChange
- PurchaseOrderNew

AdvanceShipNotice

Use this transaction to retrieve new advance ship notices. You can retrieve new advance shipment notices by Customer Name, Picking Batch Number, and Pick Slip number range. This transaction only retrieves ship notices that fit the above criteria and have not been marked as printed. The acknowledgement SQL marks the specified shipment notice as printed so that the functional query will not retrieve it a second time.

Functional Query

Inputs

The inputs are used to limit what Purchase Orders you want to retrieve. These SQL inputs are exposed as service inputs.

Item Name	Description
PICKING_BATCH	Advance shipment notices for this batch.
PICK_SLIP_NUMBER_FROM	Beginning pick slip number (inclusive).
PICK_SLIP_NUMBER_TO	Ending pick slip number (inclusive).
CUSTOMER_NAME	Advance shipment notices for this customer.

Outputs

The output contains all advance shipment notices returned by the query. Each shipment notice consists of 1 header with 1 to N lines and each line contains 1 to N shipment records. The result looks like the following:

```
Headers[0]
    Lines[0]
        Shipments[0]
        …One Shipments record for each shipment record
     …One Lines record each line item
… One Headers record for each Purchase Order
```

Header Details

A header consists of the following fields. The content of most fields can be determined by their name. Where necessary, a description is provided.

Item Name and Description (as needed)
ORG_ID
TRANSACTION_DATE
WAY_BILL_NUMBER
PICK_SLIP_NUMBER
DATE_SHIPPED
SHIP_PARTIAL_FLAG
PICKING_BATCH
NUMBER_OF_PACKAGES
FREIGHT_TERMS_CODE

Item Name and Description (as needed)
FOB_CODE
SHIPPING_METHOD_CODE
TOTAL_WEIGHT
WEIGHT_UOM
PURCHASE_ORDER_NUMBER
SALES_ORDER_NUMBER
SALES_ORDER_DATE
DATE_REQUESTED_CURRENT
ORDER_TYPE
ORDER_CATEGORY
CURRENCY_CODE
PAYMENT_TERMS
CUSTOMER_NUMBER
WAREHOUSE_LOCATION_ID
WAREHOUSE_LOCATION_CODE
WAREHOUSE_NAME
WAREHOUSE_ADDRESS1
WAREHOUSE_CITY
WAREHOUSE_POSTAL_CODE
WAREHOUSE_COUNTRY
WAREHOUSE_STATE
WAREHOUSE_TELEPHONE
SHIP_TO_ADDRESS_ID
SHIP_TO_LOCATION_CODE
CUSTOMER_NAME
SHIP_TO_ADDRESS1
SHIP_TO_CITY
SHIP_TO_POSTAL_CODE
SHIP_TO_COUNTRY
SHIP_TO_STATE

Item Name and Description (as needed)
INVOICE_TO_ADDRESS_ID
INVOICE_TO_ADDRESS1
INVOICE_TO_CITY
INVOICE_TO_POSTAL_CODE
INVOICE_TO_COUNTRY
INVOICE_TO_STATE
INVOICE_TO_CONTACT_LAST_NAME
INVOICE_TO_CONTACT_FIRST_NAME
ORDER_BY_CONTACT_LAST_NAME
ORDER_BY_CONTACT_FIRST_NAME
CUSTOMER_ATTRIBUTE15
PICKING_HEADER_ID
SHIP_TO_SITE_USE_CODE

Line Details

Line Item details consist of the following fields:

Item Name and Description (as needed)
ORG_ID
SHIPMENT_LINE_NUMBER
SALES_ORDER_LINE_NUMBER
PART_NUMBER
ITEM_TYPE_CODE
SELLING_PRICE
REQUESTED_DATE
PROMISE_DATE
SHIPMENT_LINE_SHIPPED_QTY
SHIPMENT_LINE_UNIT_OF_MEASURE
PRODUCT_DESCRIPTION
ATO_FLAG
INCLUDED_ITEM_FLAG

Item Name and Description (as needed)
OPTION_FLAG
SHIP_MODEL_COMPLETE_FLAG
LIST_PRICE
SO_LINE_DETAILS_ORDERED_QTY
SO_LINE_DETAILS_UOM
PICKING_LINE_SEQUENCE
SHIP_CONFIRMED_DATE
SHIPMENT_PRIORITY
COMPONENT_CODE
COMPONENT_RATIO
LINE_PART_ATTRIBUTE14
LINE_PART_ATTRIBUTE15
PICKING_HEADER_ID
SHIPMENTS Description: Start of shipment record.

Shipment Details

Shipment details consist of the following fields:

Item Name and Description (as needed)
SHIP_LINE_DETAIL_NUMBER
SHIP_LINE_DETAIL_SHIP_QTY
SERIAL_NUMBER
CUSTOMER_REQUESTED_LOT_FLAG
REVISION
SUBINVENTORY
PICKING_LINE_ID

Acknowledgment SQL

Updates the specified pick slip number to show it has been processed. Marks the Sales Order as Send and updates the Ship Notice Sent flag to "Y". Updates the Ship Notice Send Date and the last update date.

Inputs

The inputs define which sales order to update.

- NHEADER_ID – The PICKING_HEADER_ID to acknowledge.

Outputs

There is no output generated by this SQL.

Usage Comments

Sales Orders that have been picked up and not shipped are retrieved.

InvoiceOutbound

Use this transaction to retrieve new invoices in your Oracle Application system. You can retrieve invoices by creation date range, organization ID, billable customer name, site use code, and transaction number. Only invoices that are completed and have the printing option turned on with a status of printing pending will be returned. The acknowledgement SQL marks the specified invoice so that the functional query will not retrieve it a second time.

Functional Query

Inputs

The inputs are used to limit what Invoices you want to retrieve. These SQL inputs are exposed as service inputs.

Item Name and Description (as needed)
ORG_ID
BILL_TO_CUSTOMER_NAME
SITE_USE_CODE
TRANSACTION_NUMBER

Item Name and Description (as needed)
CREATION_DATE_FROM Description: Inclusive beginning date for CREATION_DATE field.
CREATION_DATE_TO Description: Inclusive ending date for CREATION_DATE field.

Outputs

The output contains all invoices returned by the query. Each invoice consists of 1 header with 1 to N lines and each line contains 1 to N line tax records. The results look like the following:

```
Header[0]
    Line[0]
        Line_Tax[0]
        …One Line_Tax record for each Line tax
     …One Line record each Invoice line item
… One Headers record for each Invoice
```

Header Details

The following table shows the details of an invoice header. The content of most fields can be determined by their name. Where necessary, a description is provided.

Item Name and Description (as needed)
ORG_ID
DOCUMENT_TYPE
INVOICE_NAME
DOCUMENT_NUMBER
TRANSACTION_DATE
BILL_TO_CUSTOMER_CODE_INT
BILL_TO_CUSTOMER_NAME
BILL_TO_ADDRESS1
BILL_TO_ADDRESS2
BILL_TO_ADDRESS3
BILL_TO_ADDRESS4
BILL_TO_CITY
BILL_TO_POSTAL_CODE

Item Name and Description (as needed)
BILL_TO_COUNTRY
BILL_TO_STATE
BILL_TO_PROVINCE
BILL_TO_COUNTY
BILL_TO_CUSTOMER_SIC_CODE
BILL_TO_CUSTOMER_SALES_CHANNEL
BILL_TO_CONTACT_LAST_NAME
BILL_TO_CONTACT_FIRST_NAME
BILL_TO_CONTACT_JOB_TITLE
BILL_TO_CUSTOMER_ATT_CATEGORY
BILL_TO_CUSTOMER_ATTRIBUTE1
BILL_TO_CUSTOMER_ATTRIBUTE2
BILL_TO_CUSTOMER_ATTRIBUTE3
BILL_TO_CUSTOMER_ATTRIBUTE4
BILL_TO_CUSTOMER_ATTRIBUTE5
BILL_TO_CUSTOMER_ATTRIBUTE6
BILL_TO_CUSTOMER_ATTRIBUTE7
BILL_TO_CUSTOMER_ATTRIBUTE8
BILL_TO_CUSTOMER_ATTRIBUTE9
BILL_TO_CUSTOMER_ATTRIBUTE10
BILL_TO_CUSTOMER_ATTRIBUTE11
BILL_TO_CUSTOMER_ATTRIBUTE12
BILL_TO_CUSTOMER_ATTRIBUTE13
BILL_TO_CUSTOMER_ATTRIBUTE14
BILL_TO_CUSTOMER_ATTRIBUTE15
SITE_USE_CODE
BILL_TO_SITE_ATT_CATEGORY
BILL_TO_SITE_ATTRIBUTE1
BILL_TO_SITE_ATTRIBUTE2
BILL_TO_SITE_ATTRIBUTE3

Item Name and Description (as needed)
BILL_TO_SITE_ATTRIBUTE4
BILL_TO_SITE_ATTRIBUTE5
BILL_TO_SITE_ATTRIBUTE6
BILL_TO_SITE_ATTRIBUTE7
BILL_TO_SITE_ATTRIBUTE8
BILL_TO_SITE_ATTRIBUTE9
BILL_TO_SITE_ATTRIBUTE10
BILL_TO_SITE_ATTRIBUTE11
BILL_TO_SITE_ATTRIBUTE12
BILL_TO_SITE_ATTRIBUTE13
BILL_TO_SITE_ATTRIBUTE14
BILL_TO_SITE_ATTRIBUTE15
SHIP_TO_CUSTOMER_CODE_INT
SHIP_TO_CUSTOMER_NAME
SHIP_TO_ADDRESS1
SHIP_TO_ADDRESS2
SHIP_TO_ADDRESS3
SHIP_TO_ADDRESS4
SHIP_TO_CITY
SHIP_TO_POSTAL_CODE
SHIP_TO_COUNTRY
SHIP_TO_STATE
SHIP_TO_PROVINCE
SHIP_TO_COUNTY
SHIP_TO_CUSTOMER_SIC_CODE
SHIP_TO_CUSTOMER_SALES_CHANNEL
SHIP_TO_CONTACT_LAST_NAME
SHIP_TO_CONTACT_FIRST_NAME
SHIP_TO_CONTACT_JOB_TITLE
SOLD_TO_CUSTOMER_CODE_INT

Item Name and Description (as needed)
SOLD_TO_CUSTOMER_NAME
SOLD_TO_ADDRESS1
SOLD_TO_ADDRESS2
SOLD_TO_ADDRESS3
SOLD_TO_ADDRESS4
SOLD_TO_CITY
SOLD_TO_POSTAL_CODE
SOLD_TO_COUNTRY
SOLD_TO_STATE
SOLD_TO_PROVINCE
SOLD_TO_COUNTY
SOLD_TO_CUSTOMER_SIC_CODE
SOLD_TO_CUSTOMER_SALES_CHANNEL
SOLD_TO_CONTACT_LAST_NAME
SOLD_TO_CONTACT_FIRST_NAME
SOLD_TO_CONTACT_JOB_TITLE
TRANSACTION_NUMBER
CREDITED_INVOICE_NUMBER
REFERENCE_INVOICE_NUMBER
PARENT_INVOICE_NUMBER
SHIPMENT_DATE
PURCHASE_ORDER_NUMBER
CREATION_DATE
PURCHASE_ORDER_REVISION_NUMBER
COMMITMENT_START_DATE
PURCHASE_ORDER_DATE
COMMITMENT_END_DATE
INV_TRANSACTION_DATE
LAST_UPDATE_DATE
DUE_DATE

Item Name and Description (as needed)
CREDIT_MEMO_REASON
TRANSMISSION_LEAD_DAYS
TRANSACTION_SOURCE
INSTALLMENT_NUMBER
SHIPMENT_WAYBILL_NUMBER
SHIP_VIA
SHIPMENT_FOB_POINT
CURRENCY_CODE
CURRENCY_EXCHANGE_RATE
BASE_CURRENCY_CODE
PAYMENT_TERM_NAME
PRIMARY_SALESREP_NAME
COMMENTS
BILL_TO_ADDRESS_ID
BILL_TO_CUSTOMER_LOCATION
BILL_TO_CUSTOMER_NUMBER
SHIP_TO_ADDRESS_ID
SHIP_TO_CUSTOMER_LOCATION
SHIP_TO_CUSTOMER_NUMBER
SOLD_TO_ADDRESS_ID
SOLD_TO_CUSTOMER_LOCATION
SOLD_TO_CUSTOMER_NUMBER
TRANSACTION_ID
PAYMENT_TERM_ID
BILL_TO_CUSTOMER_ID
BILL_TO_SITE_USE_ID
SHIP_TO_CUSTOMER_ID
SHIP_TO_SITE_USE_ID
SOLD_TO_CUSTOMER_ID
SOLD_TO_SITE_USE_ID

Item Name and Description (as needed)
TERM_DUE_CUTOFF_DAY
TERM_DUE_DATE
TERM_DUE_DAYS
TERM_DUE_DAY_OF_MONTH
TERM_DUE_MONTHS_FORWARD
TERM_DUE_PERCENT
REMIT_TO_ADDRESS1
REMIT_TO_ADDRESS2
REMIT_TO_ADDRESS3
REMIT_TO_ADDRESS4
REMIT_TO_CITY
REMIT_TO_COUNTY
REMIT_TO_STATE
REMIT_TO_PROVINCE
REMIT_TO_COUNTRY
REMIT_TO_CODE_INT
REMIT_TO_POSTAL_CODE
MULTIPLE_INSTALLMENTS_FLAG
MAXIMUM_INSTALLMENT_NUMBER
AMOUNT_TAX_DUE
AMOUNT_CHARGES_DUE
AMOUNT_FREIGHT_DUE
AMOUNT_LINE_INVITEMS_DUE
TOTAL_AMOUNT_DUE
DISCOUNT_PERCENT1
DISCOUNT_DAYS1
DISCOUNT_DATE1
DISCOUNT_DAY_OF_MONTH1
DISCOUNT_MONTHS_FORWARD1
DISCOUNT_PERCENT2

Item Name and Description (as needed)
DISCOUNT_DAYS2
DISCOUNT_DATE2
DISCOUNT_DAY_OF_MONTH2
DISCOUNT_MONTHS_FORWARD2
DISCOUNT_PERCENT3
DISCOUNT_DAYS3
DISCOUNT_DATE3
DISCOUNT_DAY_OF_MONTH3
DISCOUNT_MONTHS_FORWARD3

Line Details

The following table shows the details of an invoice line item. The content of most fields can be determined by their names. Where necessary, a description is provided.

Item Name and Description (as needed)
ORG_ID
TRANSACTION_ID
LINE_NUMBER
SALES_ORDER_NUMBER
SALES_ORDER_REVISION_NUMBER
SALES_ORDER_LINE_NUMBER
SALES_ORDER_DATE
SALES_CHANNEL
ITEM_ID
CUSTOMER_ITEM_NUMBER
CUSTOMER_ITEM_DESC
ITEM_DESCRIPTION
UOM_CODE
ORDERED_QUANTITY
QUANTITY
UNIT_STANDARD_PRICE

Item Name and Description (as needed)
UNIT_SELLING_PRICE
LINE_AMOUNT
CREDIT_MEMO_REASON
CREDITED_LINE_NUMBER
SHIP_ORDER_STATUS_INT
TRANSACTION_REFERENCE_KEY
INTERFACE_LINE_CATEGORY
INTERFACE_LINE_ATTRIBUTE1
INTERFACE_LINE_ATTRIBUTE2
INTERFACE_LINE_ATTRIBUTE3
INTERFACE_LINE_ATTRIBUTE4
INTERFACE_LINE_ATTRIBUTE5
INTERFACE_LINE_ATTRIBUTE6
INTERFACE_LINE_ATTRIBUTE7
INTERFACE_LINE_ATTRIBUTE8
INTERFACE_LINE_ATTRIBUTE9
INTERFACE_LINE_ATTRIBUTE10
INTERFACE_LINE_ATTRIBUTE11
INTERFACE_LINE_ATTRIBUTE12
INTERFACE_LINE_ATTRIBUTE13
INTERFACE_LINE_ATTRIBUTE14
INTERFACE_LINE_ATTRIBUTE15
LINE_ATTRIBUTE_CATEGORY
LINE_ATTRIBUTE1
LINE_ATTRIBUTE2
LINE_ATTRIBUTE3
LINE_ATTRIBUTE4
LINE_ATTRIBUTE5
LINE_ATTRIBUTE6
LINE_ATTRIBUTE7

Item Name and Description (as needed)
LINE_ATTRIBUTE8
LINE_ATTRIBUTE9
LINE_ATTRIBUTE10
LINE_ATTRIBUTE11
LINE_ATTRIBUTE12
LINE_ATTRIBUTE13
LINE_ATTRIBUTE14
LINE_ATTRIBUTE15

Line_Tax Details

The following table shows the details of an invoice line item tax record. The content of most fields can be determined by their names. Where necessary, a description is provided.

Item Name and Description (as needed)	Description
ORG_ID	
TRANSACTION_ID	
LINE_NUMBER	
CUSTOMER_TRX_LINE_ID	
TAX_LINE_NUMBER	
LINE_TYPE	
TAX_AMOUNT	
TAX_RATE	
TAX_PRECEDENCE	
TAX_CODE	
LINK_TO_CUST_TRX_LINE_ID	
TAX_ATTRIBUTE_CATEGORY	
TAX_ATTRIBUTE1	
TAX_ATTRIBUTE2	
TAX_ATTRIBUTE3	
TAX_ATTRIBUTE4	
TAX_ATTRIBUTE5	

Item Name and Description (as needed)	Description
TAX_ATTRIBUTE6	
TAX_ATTRIBUTE7	
TAX_ATTRIBUTE8	
TAX_ATTRIBUTE9	
TAX_ATTRIBUTE10	
TAX_ATTRIBUTE11	
TAX_ATTRIBUTE12	
TAX_ATTRIBUTE13	
TAX_ATTRIBUTE14	
TAX_ATTRIBUTE15	
TAX_EXEMPT_FLAG	
TAX_EXEMPT_NUMBER	
TAX_EXEMPT_REASON_CODE	
TAX_TYPE	
DESCRIPTION	
LOCATION	
TAX_CLASSIFICATION	
VAT_TRANSACTION_TYPE	
VAT_TAX_ATTRIBUTE_CATEGORY	
VAT_TAX_ATTRIBUTE1	
VAT_TAX_ATTRIBUTE2	
VAT_TAX_ATTRIBUTE3	
VAT_TAX_ATTRIBUTE4	
VAT_TAX_ATTRIBUTE5	
VAT_TAX_ATTRIBUTE6	
VAT_TAX_ATTRIBUTE7	
VAT_TAX_ATTRIBUTE8	
VAT_TAX_ATTRIBUTE9	
VAT_TAX_ATTRIBUTE10	
VAT_TAX_ATTRIBUTE11	

Item Name and Description (as needed)	Description
VAT_TAX_ATTRIBUTE12	
VAT_TAX_ATTRIBUTE13	
VAT_TAX_ATTRIBUTE14	
VAT_TAX_ATTRIBUTE15	

Acknowledgment SQL

Updates the specified invoice to show it has been processed and printed.

Inputs

The inputs define which PO to update.

- DOCUMENT_TYPE
- TRANSACTION_ID
- INSTALLMENT_NUMBER
- MULTIPLE_INSTALLMENTS_FLAG
- MAXIMUM_INSTALLMENT_NUMBER

Outputs

There is no output generated by this SQL.

Usage Comments

Invoices are retrieved using the `TRANSACTION_NUMBER` field, but they are acknowledged using the `TRANSACTION_ID` field.

PurchaseOrderChange

Use his transaction to retrieve revised purchase orders. You can retrieve new purchase orders by the creation date range, supplier, vendor, PO type, and PO number range. This transaction only retrieves Purchase Orders that fit the above criteria and have already been printed or marked as processed. The acknowledgement SQL marks the specified PO so that it will not be retrieved by the functional query a second time.

Functional Query

Inputs

Same as for the PurchaseOrderNew transaction.

Outputs

Same as for the PurchaseOrderNew transaction, except for the differences noted below.

Header Details

Does not have a `TRANSACTION_DATE` field.

Line Details

Does not have an `ORG_ID` field.

Shipment Details

Does not have an `ORG_ID` field.

Contains a `QUANTITY_PENDING` field.

Acknowledgment SQL

Same as for the PurchaseOrderNew transaction.

Inputs

Same as for the PurchaseOrderNew transaction.

Outputs

Same as for the PurchaseOrderNew transaction.

Usage Comments

- Purchase Orders that have been approved and previously printed are retrieved.
- You can retrieve only BLANKET and STANDARD Purchase Orders Types. To retrieve a STANDARD purchase order, specify a DOCUMENT_TYPE of STANDARD. To retrieve the purchase agreements, specify a DOCUMENT_TYPE of BLANKET. To retrieve releases against a purchase agreement, specify a DOCUMENT_TYPE of RELEASE.

PurchaseOrderNew

Use this transaction to retrieve new purchase orders entered into your system. You can retrieve new purchases by creation date range, supplier, vendor, PO type, and PO number range. This transaction only retrieves Purchase Orders that fit the above criteria and have not been marked as printed. The acknowledgement SQL marks the specified PO as printed so that the functional query will not retrieve it a second time.

Functional Query

Inputs

The inputs are used to limit what Purchase Orders you want to retrieve. These SQL inputs are exposed as service inputs.

Item Name	Description
CREATE_DATE_FROM	Inclusive beginning date for PO CREATION_DATE field.
CREATE_DATE_TO	Inclusive ending date for PO CREATION_DATE field.
SUPPLIER_NUMBER	Used to limit new purchase orders retrieved to this supplier.
VENDOR_SITE_ID	Used to limit new purchase orders retrieved to this vendor site.
DOCUMENT_TYPE	Used to limit new purchase orders retrieved to this document type (STANDARD, BLANKET).
PO_NUMBER_FROM	Inclusive beginning PO number.
PO_NUMBER_TO	Inclusive ending PO number.

Outputs

The output contains all purchase orders returned by the query. Each purchase order consists of 1 header with 1 to N lines and each line contains 1 to N shipment records. The result looks like the following:

```
Headers[0]
    Lines[0]
        Shipments[0]
        …One Shipments record for each shipment record
     …One Lines record each line item
… One Headers record for each Purchase Order
```

Header Details

A header consists of the following fields. The content of most fields can be determined by their name. Where necessary, a description is provided.

Item Name and Description (as needed)
ORG_ID
DOCUMENT_TYPE Description: STANDARD, BLANKET, or RELEASE.
DOCUMENT_CODE
TRANSACTION_DATE
PO_NUMBER
POR_RELEASE_ID Description: Purchase order release information.
POR_RELEASE_NUM
POR_RELEASE_DATE
CREATION_DATE
REVISION_NUM
REVISED_DATE
COMMENTS
PO_TYPE Description: STANDARD or BLANKET.
PAYMENT_TERMS
CURRENCY_CODE
CURRENCY_RATE

Item Name and Description (as needed)
SHIP_VIA
FOB_CODE
FREIGHT_TERMS
CANCEL_FLAG
ACCEPTANCE_REQUIRED_FLAG
ACCEPTANCE_DUE_DATE
CONFIRMING_ORDER_FLAG
BLANKET_START_DATE
BLANKET_END_DATE
BLANKET_TOTAL_AMOUNT
SUPPLIER_NUMBER
SUPPLIER_NAME
NOTE_TO_VENDOR
PO_ATTRIBUTE_CATEGORY Description: Purchase order attributes.
PO_ATTRIBUTE1
PO_ATTRIBUTE2
PO_ATTRIBUTE3
PO_ATTRIBUTE4
PO_ATTRIBUTE5
PO_ATTRIBUTE6
PO_ATTRIBUTE7
PO_ATTRIBUTE8
PO_ATTRIBUTE9
PO_ATTRIBUTE10
PO_ATTRIBUTE11
PO_ATTRIBUTE12
PO_ATTRIBUTE13
PO_ATTRIBUTE14
PO_ATTRIBUTE15

Item Name and Description (as needed)
SU_ATTRIBUTE_CATEGORY Description: Supplier attribute information.
SU_ATTRIBUTE1
SU_ATTRIBUTE2
SU_ATTRIBUTE3
SU_ATTRIBUTE4
SU_ATTRIBUTE5
SU_ATTRIBUTE6
SU_ATTRIBUTE7
SU_ATTRIBUTE8
SU_ATTRIBUTE9
SU_ATTRIBUTE10
SU_ATTRIBUTE11
SU_ATTRIBUTE12
SU_ATTRIBUTE13
SU_ATTRIBUTE14
SU_ATTRIBUTE15
SS_ATTRIBUTE_CATEGORY Description: Supplier site attributes.
SS_ATTRIBUTE1
SS_ATTRIBUTE2
SS_ATTRIBUTE3
SS_ATTRIBUTE4
SS_ATTRIBUTE5
SS_ATTRIBUTE6
SS_ATTRIBUTE7
SS_ATTRIBUTE8
SS_ATTRIBUTE9
SS_ATTRIBUTE10
SS_ATTRIBUTE11

Item Name and Description (as needed)
SS_ATTRIBUTE12
SS_ATTRIBUTE13
SS_ATTRIBUTE14
SS_ATTRIBUTE15
CUSTOMER_NUMBER Description: Vendor customer number.
VENDOR_SITE_ID
SU_ADDRESS_LINE1
SU_ADDRESS_LINE2
SU_ADDRESS_LINE3
SU_ADDRESS_LINE4
SU_CITY
SU_ZIP
SU_COUNTRY
SU_STATE
SU_PROVINCE
SU_AREA_CODE
SU_PHONE
SU_FAX_AREA_CODE
SU_FAX
SU_TELEX
CN_LAST_NAME Description: Vendor contact information.
CN_FIRST_NAME
CN_AREA_CODE
CN_PHONE
ST_LOCATION_ID Description: Ship-to information.
ST_CONTACT_LAST_NAME
ST_CONTACT_FIRST_NAME

Item Name and Description (as needed)
ST_NAME
ST_ADDRESS_LINE1
ST_ADDRESS_LINE2
ST_ADDRESS_LINE3
ST_CITY
ST_POSTAL_CODE
ST_COUNTRY
ST_REGION1
ST_REGION2
ST_REGION3
ST_PHONE1
ST_PHONE2
ST_PHONE3
BT_LOCATION_ID Description: Bill-to information.
BT_CONTACT_LAST_NAME
BT_CONTACT_FIRST_NAME
BT_NAME
BT_ADDRESS_LINE1
BT_ADDRESS_LINE2
BT_ADDRESS_LINE3
BT_CITY
BT_POSTAL_CODE
BT_COUNTRY
BT_REGION1
BT_REGION2
BT_REGION3
BT_PHONE1
BT_PHONE2
BT_PHONE3

Item Name and Description (as needed)
BUYER_LAST_NAME
BUYER_FIRST_NAME
BUYER_EMAIL_ADDRESS
BUYER_WORK_TELEPHONE
BUYER_MISC_TELEPHONE1
BUYER_MISC_TELEPHONE2
BUYER_MISC_TELEPHONE3
PO_HEADER_ID
FSP_INVENTORY_ORGANIZATION_ID Description: Financial system parameter inventory organization ID.

Line Details

Line Item details consist of the following fields:

Item Name and Description (as needed)
ORG_ID
LINE_NUM
QUANTITY
UOM_CODE Description: Unit of measure code.
ITEM_ID Definition: Inventory item ID.
ITEM_REVISION
VENDOR_PRODUCT_NUMBER
UNIT_PRICE
ITEM_DESCRIPTION
PO_NUMBER
VENDOR_QUOTE_NUMBER
CANCEL_FLAG
CANCEL_DATE
QUANTITY_COMMITTED

Item Name and Description (as needed)
COMMITTED_AMOUNT
LIST_PRICE_PER_UNIT
MARKET_PRICE
NOT_TO_EXCEED_PRICE
NEGOTIATED_BY_PREPARER_FLAG
TAXABLE_FLAG
TRANSACTION_REASON_CODE
LINE_TYPE
HAZARD_CLASS
UN_NUMBER
UN_DESCRIPTION
NOTE_TO_VENDOR
POL_ATTRIBUTE_CATEGORY Definition: Purchase order line attributes.
POL_ATTRIBUTE1
POL_ATTRIBUTE2
POL_ATTRIBUTE3
POL_ATTRIBUTE4
POL_ATTRIBUTE5
POL_ATTRIBUTE6
POL_ATTRIBUTE7
POL_ATTRIBUTE8
POL_ATTRIBUTE9
POL_ATTRIBUTE10
POL_ATTRIBUTE11
POL_ATTRIBUTE12
POL_ATTRIBUTE13
POL_ATTRIBUTE14
POL_ATTRIBUTE15
MTL_ORGANIZATION_ID

Item Name and Description (as needed)
LP_ATTRIBUTE_CATEGORY Description: Material item attributes.
LP_ATTRIBUTE1
LP_ATTRIBUTE2
LP_ATTRIBUTE3
LP_ATTRIBUTE4
LP_ATTRIBUTE5
LP_ATTRIBUTE6
LP_ATTRIBUTE7
LP_ATTRIBUTE8
LP_ATTRIBUTE9
LP_ATTRIBUTE10
LP_ATTRIBUTE11
LP_ATTRIBUTE12
LP_ATTRIBUTE13
LP_ATTRIBUTE14
LP_ATTRIBUTE15
LP_SEGMENT1
LP_SEGMENT2
LP_SEGMENT3
LP_SEGMENT4
LP_SEGMENT5
LP_SEGMENT6
LP_SEGMENT7
LP_SEGMENT8
LP_SEGMENT9
LP_SEGMENT10
LP_SEGMENT11
LP_SEGMENT12
LP_SEGMENT13

Item Name and Description (as needed)
LP_SEGMENT14
LP_SEGMENT15
LP_SEGMENT16
LP_SEGMENT17
LP_SEGMENT18
LP_SEGMENT19
LP_SEGMENT20
POR_RELEASE_ID
POR_RELEASE_NUM
POR_SOURCE_LINE_NUM
PO_HEADER_ID Description: Shows relationship to PO header.
PO_LINE_ID
SHIPMENT_TYPE
SHIPMENTS Definition: Start of shipment record.

Shipment Details

Shipment details consist of the following fields:

Item Name and Description (as needed)
ORG_ID
SHIPMENT_NUMBER
QUANTITY_ORIGINAL
UOM_CODE Description: Unit of measure code.
SHIPMENT_NEED_BY_DATE
SHIPMENT_PROMISED_DATE
SHIPMENT_LAST_ACCEPTABLE_DATE
QUANTITY_CANCELLED
QUANTITY_RECEIVED

Item Name and Description (as needed)
PRICE_OVERRIDE
CANCELLED_FLAG
CANCELLED_DATE
SHIP_VIA
FOB_CODE
FREIGHT_TERMS
TAXABLE_FLAG
SHIPMENT_ATTRIBUTE_CATEGORY
SHIPMENT_ATTRIBUTE1
SHIPMENT_ATTRIBUTE2
SHIPMENT_ATTRIBUTE3
SHIPMENT_ATTRIBUTE4
SHIPMENT_ATTRIBUTE5
SHIPMENT_ATTRIBUTE6
SHIPMENT_ATTRIBUTE7
SHIPMENT_ATTRIBUTE8
SHIPMENT_ATTRIBUTE9
SHIPMENT_ATTRIBUTE10
SHIPMENT_ATTRIBUTE11
SHIPMENT_ATTRIBUTE12
SHIPMENT_ATTRIBUTE13
SHIPMENT_ATTRIBUTE14
SHIPMENT_ATTRIBUTE15
SHIP_TO_LOCATION_ID
SHIP_TO_LOCATION_CODE
SHIP_TO_CONTACT_LAST_NAME
SHIP_TO_CONTACT_FIRST_NAME
SHIP_TO_ADDRESS_LINE_1
SHIP_TO_ADDRESS_LINE_2
SHIP_TO_ADDRESS_LINE_3

Item Name and Description (as needed)
SHIP_TO_CITY
SHIP_TO_POSTAL_CODE
SHIP_TO_COUNTRY
SHIP_TO_REGION_1
SHIP_TO_REGION_2
SHIP_TO_REGION_3
SHIP_TO_PHONE1
SHIP_TO_PHONE2
SHIP_TO_PHONE3
PO_HEADER_ID Description: Shows relationship to PO header.
PO_LINE_ID Description: Shows relationship to PO line.
POR_RELEASE_ID
POR_RELEASE_NUM
LINE_LOCATION_ID

Acknowledgment SQL

Updates the specified purchase order to show it has been processed. Marks the PO as printed and updates the print date, edi_processed_flag, and the last update date. It also performs the necessary archive procedures. It updates and archives a single PO at a time.

Inputs

The inputs define which PO to update are:

- ORG_ID
- DOCUMENT_TYPE
- PO_NUMBER
- PO_TYPE
- POR_RELEASE_NUMBER

Outputs

There is no output generated by this SQL.

Usage Comments

- Purchase Orders that have been approved and not printed are retrieved.
- Purchase Orders that are Cancelled or on Hold are not retrieved.
- Only BLANKET and STANDARD Purchase Orders Types can be retrieved.

CHAPTER 7

Oracle Applications Adapter Built-In Utility Services

Built-In Utility Services

The following sections describe the built-in services you can use in the WmOAAdapter package.

pub.adapter.wmoa:deleteAllTransactions

This service allows you to delete all transaction definitions at one time using the Developer, rather than deleting each transaction definition individually using the Server Administrator.

Input Parameters

OARelease

Value. The Oracle Applications release number of the transaction definitions to be deleted.

Valid values: 107SC, 107NCA, 11, 11i

Output Parameters

deleteStatus

fOARelease The Oracle Applications release number.

transactionName The name of the transaction definition.

message Informational message.

pub.adapter.wmoa:importAllTransactions

This service allows you to import all of the transaction definitions into the repository for a particular release of Oracle Applications at one time using the Developer, rather than importing the transaction definitions individually using the Server Administrator.

There are no input parameters. The service imports all .txp files located in the *IntegrationServer_Directory*\packages\WmOAAdapter\exchange directory into the repository.

Output Parameters

importStatus

success Either true or false.

message Informational message.

fileTranName The name of the transport file (.txp) of the transaction definition.

importTranName Transaction definition name.

Index